Faceauth:
Sistema de Autenticação Facial IoT

Gabriel Alexandre Carvalho

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, 2024

Gabriel Alexandre Carvalho

Faceauth

Monografia apresentada ao curso Engenharia da Computação
do Centro de Informática, da Universidade Federal da Paraíba,
como requisito para a obtenção do grau de Bacharel em título

Orientadora:Verônica Maria Lima Silva

Novembro de 2024

Catalogação na publicação
Seção de Catalogação e Classificação

C331s Carvalho, Gabriel Alexandre.
Sistema de autenticação facial IoT / Gabriel Alexandre Carvalho. - João Pessoa, 2024.
47 f. : il.

Orientação: Verônica Maria Lima Silva.
TCC (Graduação) - UFPB/CI.

1. Autenticação Facial. 2. Internet das Coisas (IoT). 3. Arquitetura web. 4. Segurança da informação. 5. Reconhecimento facial. I. Silva, Verônica Maria Lima. II. Título.

UFPB/CI CDU 004.056

Elaborado por Michelle de Kássia Fonseca Barbosa - CRB-738

CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Engenharia da Computaçãointitulado ***Faceauth*** de autoria de Gabriel Alexandre Carvalho, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. Alisson Vasconcelos De Brito
CI/UFPB

Prof. Dr. Ewerton Monteiro Salvador
CI/UFPB

Coordenador(a) do Departamento Departamento de Informática
Josilene Aires Moreira
CI/UFPB

João Pessoa, 15 de novembro de 2024

Centro de Informática, Universidade Federal da Paraíba
Rua dos Escoteiros, Mangabeira VII, João Pessoa, Paraíba, Brasil CEP: 58058-600
Fone: +55 (83) 3216 7093 / Fax: +55 (83) 3216 7117

AGRADECIMENTOS

Primeiramente, agradeço a Deus por ter me dado força, saúde e sabedoria ao longo dessa jornada. Sem Sua presença constante em minha vida, eu não teria conseguido superar os desafios e chegar até aqui.

Agradeço profundamente aos meus pais e irmãos, cuja dedicação, amor incondicional e paciência foram fundamentais para que eu chegasse até aqui. Eles sempre acreditaram em mim, e sou imensamente grato por todo o apoio e incentivo que recebi. Aos meus amigos, em especial Anderson e Laura, que estiveram comigo durante toda a graduação. Anderson, sua ajuda foi essencial para que eu pudesse superar muitos desafios ao longo da graduação. Laura, juntos enfrentamos muitos desafios, mas a sua presença ao longo dessa jornada foi essencial e tornou cada obstáculo mais fácil de superar. Agradeço muito por ter contado com a sua ajuda em cada etapa. Agradeço profundamente por compartilharmos tantos momentos e aprendizados juntos. Foi uma caminhada árdua, cheia de desafios, mas com a ajuda de todos vocês, consegui superar cada um deles e concluir essa etapa tão importante da minha vida.

RESUMO

O crescente uso de dispositivos de Internet das Coisas (IoT) em aplicações críticas requer métodos de autenticação que sejam robustos e acessíveis, considerando as limitações de processamento e armazenamento desses dispositivos. O objetivo deste trabalho é desenvolver um sistema de autenticação facial integrado a uma arquitetura IoT. A metodologia inclui a utilização da ESP32-CAM para a captura de imagens faciais, o Supabase para o gerenciamento de dados e autenticação, e o Next.js como interface de front-end. O sistema de autenticação facial apresentou uma acurácia média de 91,46%, com 95% de precisão em distâncias de 30 cm e 60 cm, caindo para 85% a 1,30 m. O tempo médio de resposta foi de 7,31 segundos para usuários cadastrados e 10,4 segundos para não cadastrados. Esses resultados sugerem que o sistema de autenticação facial é viável para aplicações práticas em ambientes IoT, o que indica sua capacidade de operar mesmo sob limitações de dispositivos com recursos restritos.

Palavras-chave: Autenticação Facial, Internet das Coisas (IoT), Arquitetura Web, Segurança da Informação, Reconhecimento Facial.

Sumário

1 INTRODUÇÃO

A autenticação facial tem se destacado como uma solução biométrica eficaz em meio ao crescimento da Internet das Coisas (IoT), fornecendo maior segurança ao substituir senhas tradicionais, que são vulneráveis a ataques como a força bruta e o uso de credenciais roubadas. Essa tecnologia utiliza características faciais únicas para autenticação e, com o apoio de algoritmos de inteligência artificial, como redes neurais, tem apresentado avanços consideráveis em precisão e eficiência. Além disso, sua aplicação em dispositivos IoT tem sido uma opção viável e leve, atendendo às limitações computacionais desses dispositivos conectados. Projeções indicam que o mercado de reconhecimento facial atingirá US$ 7 bilhões até 2024, impulsionado pela demanda crescente em setores como automação residencial e segurança industrial [6, 7, 8].

O desenvolvimento de sistemas de autenticação facial em dispositivos IoT, contudo, enfrenta barreiras tecnológicas, como a limitação de recursos computacionais e a necessidade de garantir a privacidade dos dados biométricos. Este trabalho visa abordar essas questões, propondo uma solução que equilibre eficiência, segurança e custo, tornando a autenticação facial acessível para um maior número de aplicações dentro do ecossistema IoT.

1.1 Definição do Problema

A crescente dependência de dispositivos conectados à Internet das Coisas (IoT) em aplicações críticas, como segurança doméstica, saúde e automação industrial, tem aumentado a necessidade de métodos robustos de autenticação. Enquanto métodos tradicionais, como senhas e tokens, apresentam vulnerabilidades que podem ser exploradas por atacantes, a autenticação biométrica, especificamente o reconhecimento facial, surge como uma solução promissora por sua capacidade de fornecer uma identificação única e menos suscetível a fraudes.

Implementar um sistema de autenticação facial eficiente em ambientes IoT apresenta desafios significativos. Um dos principais problemas é embarcar algoritmos de reconhecimento facial em dispositivos IoT, que frequentemente possuem recursos computacionais limitados e precisam operar com baixo consumo de energia. Esses dispositivos não têm a capacidade de processar os dados de imagem necessários para algoritmos complexos, como redes neurais profundas, que são comumente utilizadas para aumentar a precisão do reconhecimento facial. Além disso, garantir a privacidade e segurança desses dados, sem comprometer a eficiência, torna-se um desafio adicional [7, 8]. Além disso, garantir a privacidade dos dados biométricos, que são altamente sensíveis, e assegurar a interoperabilidade entre diferentes dispositivos e plataformas são questões complexas que precisam ser abordadas. Este trabalho busca investigar esses problemas desenvolvendo

uma solução de autenticação facial que seja ao mesmo tempo eficiente, segura e de baixo custo, viabilizando seu uso em aplicações de IoT.

O sistema desenvolvido consiste em uma solução de autenticação facial para ambientes IoT, utilizando uma arquitetura modular que integra diversas tecnologias. A ESP32-CAM realiza a captura de imagens faciais, que são enviadas via protocolo MQTT para o backend, onde são processadas por meio do face-api.js. O backend utiliza o Supabase, que gerencia a autenticação, armazenamento e banco de dados. O Next.js funciona como interface de front-end, permitindo a interação dos usuários com o sistema. A inteligência artificial implementada realiza o reconhecimento facial utilizando modelos de redes neurais, que comparam as características faciais com as imagens armazenadas, assegurando a autenticação. A comunicação entre os módulos é projetada para otimizar o consumo de recursos, atendendo às limitações dos dispositivos IoT.

1.2 Premissas e Hipóteses

Para o desenvolvimento deste estudo, partimos de algumas premissas fundamentais. A primeira é que os dispositivos IoT utilizados no experimento terão limitações de processamento e armazenamento, o que implica a necessidade de algoritmos de autenticação facial que sejam otimizados para operar em hardware restrito. Outra premissa é que a comunicação entre os dispositivos IoT e a infraestrutura web ocorrerá através de protocolos de comunicação seguros, como o MQTT, que garantem a integridade e confidencialidade dos dados transmitidos.

Com base nessas premissas, o trabalho se apoia nas seguintes hipóteses:

1. É possível desenvolver uma solução de autenticação facial eficiente e de baixo custo para ambientes IoT, utilizando algoritmos de reconhecimento facial otimizados para dispositivos com recursos limitados.

2. A integração de um sistema de autenticação facial com uma arquitetura IoT pode ser realizada de maneira segura, garantindo a proteção dos dados biométricos e a privacidade dos usuários.

3. O sistema proposto terá um desempenho comparável ou superior a soluções existentes, em termos de precisão e tempo de resposta, mesmo em um ambiente de baixo custo e com limitações de hardware.

Estas hipóteses serão testadas ao longo do desenvolvimento e da avaliação do sistema proposto, com o intuito de validar a viabilidade e a eficácia da solução sugerida.

1.3 Objetivo geral

O objetivo geral deste trabalho é desenvolver e avaliar um sistema de autenticação facial integrado a uma arquitetura IoT, com ênfase na criação de uma solução de baixo custo que mantenha padrões adequados de segurança e eficiência.

1.4 Objetivos específicos

1. Realizar uma revisão das principais tecnologias e algoritmos utilizados em sistemas de reconhecimento facial e sua aplicabilidade em dispositivos IoT.
2. Projetar e implementar um protótipo de sistema de autenticação facial que se integre eficientemente com uma infraestrutura IoT.
3. Avaliar o desempenho do sistema proposto em termos de precisão e tempo de resposta.
4. Comparar a solução desenvolvida com alternativas existentes, destacando suas vantagens e limitações.
5. Propor melhorias e direções futuras para a expansão e aprimoramento da solução.

1.5 Estrutura do Trabalho

Este trabalho está organizado em seis capítulos principais. O Capítulo 1 apresenta a introdução, incluindo a definição do problema, premissas e hipóteses, objetivos e a estrutura do trabalho. O Capítulo 2 aborda os conceitos gerais relacionados à autenticação facial, IoT e arquitetura web, fornecendo o embasamento teórico necessário para o desenvolvimento do estudo. No Capítulo 3, são detalhadas a metodologia utilizada, as ferramentas escolhidas e o desenho do experimento. O Capítulo 4 apresenta os resultados obtidos, seguidos por uma análise crítica desses resultados. O Capítulo 5 traz as conclusões, onde são discutidas as contribuições do estudo e propostas de trabalhos futuros. Por fim, o Capítulo 6 lista as referências bibliográficas utilizadas ao longo do trabalho.

2 CONCEITOS GERAIS E REVISÃO DA LITERATURA

Nesta seção, serão apresentados os conceitos fundamentais relacionados ao sistema de autenticação facial integrado a arquiteturas de Internet das Coisas (IoT) e web. O objetivo é fornecer uma base teórica, abordando o estado da arte e as tecnologias mais relevantes, com foco em como essas inovações se aplicam ao problema estudado.

2.1 Autenticação Facial

1. **Tecnologias e Algoritmos**

   Na autenticação facial para ambientes que combinam IoT e arquitetura web, são utilizadas tecnologias como a ESP32-CAM e a biblioteca face-api.js no lado do servidor para realizar o processamento e reconhecimento facial.

   (a) **Tecnologias:**

      i. **ESP32-CAM:** O ESP32-CAM é um módulo de câmera que utiliza o microcontrolador ESP32, equipado com conectividade Wi-Fi e Bluetooth, além de suporte para outros protocolos, tornando-o adequado para uma ampla variedade de aplicações de IoT e visão computacional. Ele é capaz de capturar imagens em resoluções variadas, de 256x256 até 1600x1200 pixels, permitindo adaptar a qualidade da imagem às necessidades do projeto. Em aplicações como autenticação facial, o ESP32-CAM atua como dispositivo de captura, enviando as imagens para um servidor backend onde ocorre o processamento necessário. Essa flexibilidade de comunicação e de resolução de imagem torna o ESP32-CAM uma opção eficiente para projetos que exigem captura e transmissão de imagens [9].

      ii. **face-api.js:** O face-api.js é uma biblioteca JavaScript de código aberto construída sobre o TensorFlow.js, desenvolvida por Vincent Mühler, que realiza reconhecimento facial utilizando técnicas de aprendizado profundo. Quando executada no servidor, a biblioteca permite a análise e o reconhecimento facial diretamente no backend, eliminando a necessidade de enviar imagens para serviços de terceiros e aumentando a segurança e a privacidade dos dados. A principal forma de uso do face-api.js é através da comparação de duas imagens: o modelo pré-treinado compara as imagens recebidas como entrada, não sendo necessário treinar um modelo com imagens de usuários específicos [10].

   (b) **Funcionamento do face-api.js:**

      i. **Modelos Utilizados:** O face-api.js emprega redes neurais convolucionais (CNNs) treinadas em grandes bases de dados de imagens faciais, como

a Labelled Faces in the Wild (LFW) e a VGGFace2. Esses modelos são capazes de realizar várias tarefas de reconhecimento facial, incluindo detecção de rostos, reconhecimento de identidade e extração de landmarks faciais (pontos de referência como olhos, nariz e boca) [11].

ii. **Processo de Reconhecimento:** O reconhecimento facial com o face-api.js envolve várias etapas, começando com a detecção de rostos na imagem, seguida pela extração de características faciais únicas para cada rosto detectado. A biblioteca utiliza modelos pré-treinados para mapear rostos em um espaço de características de alta dimensão, permitindo comparações precisas entre diferentes imagens faciais.

iii. **Treinamento e Precisão:** Os modelos no face-api.js foram treinados usando técnicas avançadas de aprendizado supervisionado, aproveitando conjuntos de dados ricos e diversificados para capturar uma ampla gama de variações faciais devido a iluminação, postura e expressão. Esses modelos são otimizados para alta precisão e são frequentemente atualizados para melhorar sua robustez em diferentes cenários.

2. **Aplicações**

(a) **Segurança e Vigilância:** A autenticação facial é amplamente empregada em sistemas de segurança e vigilância para identificar e monitorar indivíduos em tempo real. Em ambientes como aeroportos e edifícios corporativos, câmeras equipadas com reconhecimento facial permitem a verificação de identidades e o controle de acesso. Exemplos incluem o uso de tecnologias semelhantes ao face-api.js para comparar imagens capturadas pelas câmeras com bancos de dados de pessoas autorizadas, emitindo alertas para atividades suspeitas [12].

(b) **Internet das Coisas (IoT):** Em sistemas IoT, a autenticação facial é usada para controlar o acesso a dispositivos e sistemas conectados. Fechaduras inteligentes e sistemas de gerenciamento de acesso empregam reconhecimento facial para permitir ou negar entrada com base na identidade do usuário. Tecnologias como a ESP32-CAM são integradas com algoritmos de reconhecimento facial para proporcionar um acesso seguro e sem fricções [13].

2.2 Internet das Coisas (IoT)

1. **Arquitetura de IoT**

A arquitetura de IoT é dividida em cinco camadas principais [14], cada uma com funções específicas:

(a) **Componentes:** Incluem dispositivos IoT, sensores, atuadores e sistemas de comunicação. Dispositivos como a ESP32-CAM capturam dados (imagens) e os enviam para processamento.

(b) **Camada de Percepção:** Responsável pela coleta de dados do ambiente por meio de sensores e dispositivos. A ESP32-CAM atua nesta camada, capturando imagens faciais e enviando-as para o backend.

(c) **Camada de Rede:** Faz a transmissão dos dados coletados para sistemas de processamento. O protocolo MQTT é utilizado para enviar as imagens da ESP32-CAM para o backend, proporcionando uma comunicação eficiente e leve entre dispositivos e servidores.

(d) **Camada de Processamento:** Realiza a análise e processamento dos dados recebidos. No backend, o processamento das imagens é feito usando bibliotecas como face-api.js para reconhecimento facial.

(e) **Camada de Aplicação:** Fornece a interface e os serviços que os usuários interagem, como aplicativos de segurança, sistemas de controle de acesso e plataformas de monitoramento.

2. **Protocolo de Comunicação**

   O protocolo de comunicação utilizado é o MQTT (Message Queuing Telemetry Transport), ele é um protocolo de comunicação leve e eficiente, ideal para redes com largura de banda limitada e alta latência. No contexto de IoT, o MQTT é utilizado para a transmissão de mensagens entre dispositivos e servidores. A ESP32-CAM utiliza MQTT para enviar imagens faciais ao backend. O backend também envia sinais de controle para a ESP32-CAM, indicando quando o dispositivo deve capturar e enviar uma nova imagem. Esses sinais garantem a sincronização e o controle eficiente da captura e envio das imagens [15].

3. **Comunicação entre Dispositivos IoT e Servidores Web**

   A comunicação entre dispositivos IoT e servidores web é gerida através do protocolo MQTT, mas os dados não são enviados em tempo real. A ESP32-CAM transmite imagens apenas quando recebe uma solicitação específica do backend. O backend envia mensagens para a ESP32-CAM solicitando o envio de imagens. Assim que a ESP32-CAM recebe essa solicitação, ela captura e envia a imagem para o backend para processamento e análise [16].

4. **Segurança**

   (a) **Segurança na arquitetura Web:** Protocolos como HTTPS garantem que a comunicação entre dispositivos e servidores seja criptografada, protegendo os

dados contra interceptações e ataques man-in-the-middle. A implementação de autenticação robusta, como OAuth2, e práticas de segurança, como a gestão segura de chaves e tokens, são cruciais para proteger tanto os dispositivos quanto os dados que trafegam na rede.

(b) **Segurança na arquitetura IoT:** Na camada de IoT, medidas adicionais incluem a autenticação de dispositivos, garantindo que apenas dispositivos autorizados possam se conectar e se comunicar com o servidor. A configuração de tópicos e permissões no MQTT deve ser feita de forma a garantir que as mensagens sejam transmitidas e recebidas de maneira segura e apropriada.

2.3 Trabalhos Relacionados e Estado da Arte

Este tópico revisa pesquisas relevantes na área de autenticação facial, com foco em sistemas que integram dispositivos IoT e técnicas avançadas de reconhecimento facial. Os estudos abordam a aplicação de diferentes tecnologias e metodologias para melhorar a segurança e a funcionalidade dos sistemas de autenticação facial.

1. **Autenticação Facial com Reconhecimento de Linguagem de Sinais Usando ESP32-CAM:** Os autores do artigo [1], Yalçın, Türkdağlı, Dalkılıç e Aydın (2023) exploram o uso da ESP32-CAM para autenticação facial combinada com reconhecimento de linguagem de sinais. O estudo destaca a integração de dispositivos IoT com inteligência artificial para aprimorar a segurança e a acessibilidade. A pesquisa detalha o processo de implementação da ESP32-CAM para capturar imagens faciais e processá-las para reconhecimento, enquanto também interpreta gestos de linguagem de sinais. O sistema proposto demonstra a eficácia de um processo de autenticação em modo duplo, que atende a usuários com deficiência auditiva e de fala, ampliando a usabilidade das tecnologias de reconhecimento facial em ambientes diversos.

2. **Sistema de Segurança Residencial com Reconhecimento Facial Baseado em Redes Neurais Convolucionais:** Os autores do artigo [2], Irjanto e Surantha (2020) apresentam um sistema de segurança residencial que utiliza redes neurais convolucionais (CNN) para reconhecimento facial. O estudo fornece insights sobre como modelos CNN podem ser treinados para identificar indivíduos com precisão em um ambiente doméstico, aprimorando a segurança ao prevenir acessos não autorizados. O artigo descreve a arquitetura do sistema, incluindo os componentes de hardware, o processo de treinamento da CNN e os métodos usados para otimizar a precisão do reconhecimento. Os resultados mostram melhorias significativas na segurança e na eficiência, reforçando o uso da inteligência artificial em sistemas de segurança residencial.

3. **Autenticação Facial Robusta Aproveitando Sensores Acústicos em Smartphones:** Os autores do artigo [3], Zhou, Xie, Zhang, Lohokare, Gao e Ye (2021) investigam uma abordagem inovadora para autenticação facial utilizando tecnologia de sensores acústicos integrados em smartphones. O estudo explora como sinais acústicos, em conjunto com o reconhecimento facial baseado em imagens, podem aprimorar a robustez dos processos de autenticação. A metodologia inclui a captura de imagens faciais enquanto sinais acústicos são utilizados para detectar a vivacidade e prevenir ataques de spoofing. Esta abordagem de dupla detecção aborda vulnerabilidades comuns em sistemas de reconhecimento facial, proporcionando um método de autenticação mais seguro e confiável para dispositivos móveis.

4. **Autenticação Facial com Mudanças de Maquiagem:** Os autores do artigo [4], Guo, Wen e Yan (2014) abordam o desafio da autenticação facial na presença de maquiagem, que pode alterar significativamente a aparência de uma pessoa e afetar a precisão do reconhecimento. Os autores propõem algoritmos avançados capazes de distinguir entre características faciais genuínas e alterações causadas por maquiagem. O estudo envolve experimentação detalhada com vários estilos de maquiagem e seu impacto em sistemas de reconhecimento. Ao aprimorar os modelos de reconhecimento facial para levar em conta essas mudanças, o artigo fornece soluções que melhoram a robustez dos sistemas de autenticação, tornando-os mais confiáveis em cenários do mundo real onde os usuários podem alterar frequentemente sua aparência.

3 METODOLOGIA

Esta seção aborda as ferramentas e tecnologias utilizadas para o desenvolvimento do sistema de autenticação facial, a arquitetura do sistema, a implementação e o funcionamento, além dos testes realizados.

3.1 Arquitetura do sistema

A Figura 1 mostra a arquitetura do sistema, que é composta por seis módulos principais: front-end, services, database, storage, IA e ESP32-CAM. O front-end é responsável pela interface de interação do usuário, permitindo o envio de solicitações ao back-end, como login e captura de imagem. Ele se comunica exclusivamente com o módulo services, que faz a intermediação entre os outros módulos. O módulo database armazena informações essenciais, como dados dos usuários e logs do sistema, recebendo e enviando dados por meio do módulo services. O módulo storage é responsável pelo armazenamento de arquivos, como as imagens dos usuários, também se comunicando apenas com o módulo services para receber as imagens e fornecer os links de acesso.

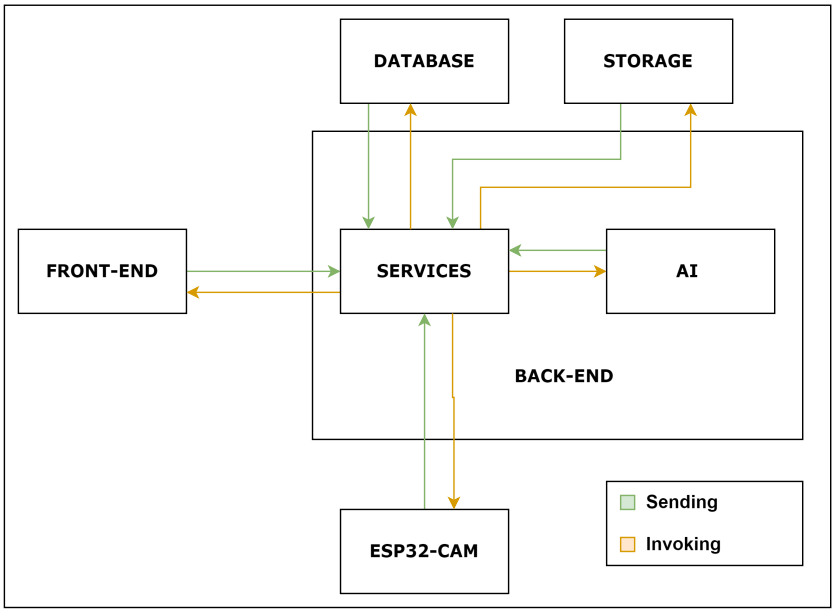

Figura 1: Arquitetura do sistema

O módulo de IA realiza o processamento das imagens capturadas, extraindo características faciais e comparando-as para validar a autenticação, enviando os resultados ao módulo services. A ESP32-CAM captura as imagens dos usuários e se conecta diretamente ao módulo services, que recebe as imagens e envia comandos para capturas adicionais. O módulo services, por sua vez, é o núcleo central do sistema, coordenando a comunicação entre todos os outros módulos por meio de APIs, além de realizar pré-processamento e gerenciamento dos dados. O back-end do sistema é formado pelo módulo services e o módulo de IA, que juntos realizam as principais operações da aplicação. Seguindo essa estrutura, o fluxo de dados do processo de autenticação pode ser representado pela Figura 2.

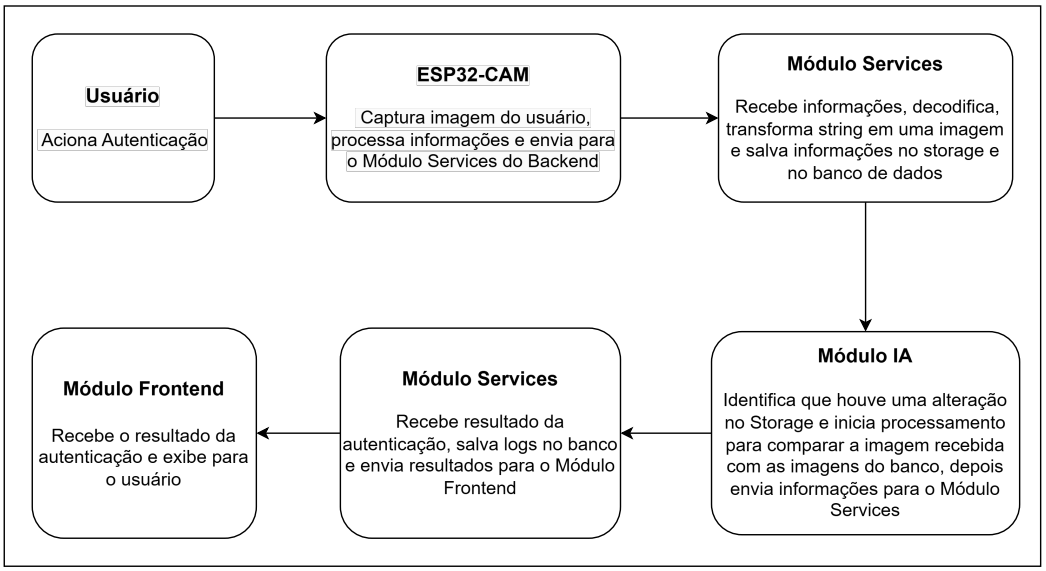

Figura 2: Fluxo de dados

3.2 Módulo IoT

O módulo IoT é composto pela ESP32-CAM, cujo objetivo principal é receber sinais de controle e enviar imagens capturadas para o módulo services da aplicação. O código utilizado para programar a ESP32-CAM foi desenvolvido em C++ na IDE Arduino. Este módulo opera seguindo uma série de rotinas, começando pela recepção de sinais de controle, passando pela captura e processamento das imagens, e finalizando com o envio dos dados via protocolo MQTT.

1. **Configuração:** O primeiro passo na configuração da ESP32-CAM envolve a definição do broker MQTT, que será responsável pela intermediação da comunicação entre o dispositivo e o módulo services. São configurados os tópicos nos quais o dispositivo irá se inscrever para receber comandos e aqueles onde ele irá publicar os dados, como as imagens capturadas. Além disso, é realizada a configuração de todas

as bibliotecas necessárias para o funcionamento do dispositivo ao longo do código. Isso inclui bibliotecas como WiFi.h para a conexão com a rede, PubSubClient.h para a comunicação via MQTT, e esp_camera.h para o controle da câmera, garantindo que todos os recursos estejam disponíveis e prontos para uso durante a execução do programa.

2. **Protocolo de Comunicação (MQTT):** O protocolo MQTT (Message Queuing Telemetry Transport) é um protocolo leve que opera sobre TCP, ideal para redes com largura de banda limitada e latência elevada, características comuns em sistemas IoT. A ESP32-CAM utiliza o protocolo MQTT para enviar dados ao módulo services devido à sua eficiência e confiabilidade no transporte de mensagens em dispositivos com recursos limitados. MQTT permite o envio de dados de forma assíncrona e usa tópicos para organizar as mensagens, proporcionando um controle eficiente da comunicação. O meio de transmissão escolhido para esse módulo é o Wi-Fi, compatível com o alcance médio da ESP32, que pode cobrir aproximadamente 50 metros em ambientes abertos. Esse alcance é adequado para aplicações em que a ESP32-CAM está fisicamente próxima ao ponto de acesso Wi-Fi. A escolha do Wi-Fi para a transmissão é uma vantagem, considerando a facilidade de implementação e a capacidade de lidar com o volume de dados da transmissão de imagens.

3. **Recepção de Sinais de Controle:** O módulo services envia sinais de controle para a ESP32-CAM, utilizando números inteiros para indicar ações específicas. O sinal "1"indica que o dispositivo deve capturar e enviar uma imagem, enquanto o sinal "0"cancela a operação. Esses sinais são recebidos via protocolo MQTT e, ao serem identificados, disparam a execução de rotinas apropriadas no dispositivo, conforme a solicitação recebida.

4. **Captura de Imagem:** A captura de imagem é realizada por meio da biblioteca esp_camera.h, que controla o hardware da câmera e retorna a imagem capturada em um array de bytes. Esse array contém os dados brutos da imagem e será processado para ser transmitido ao sistema. A configuração da câmera foi ajustada para capturar imagens em uma resolução de 256x256 pixels, visando otimizar o uso dos recursos de processamento e transmissão, o que é essencial para dispositivos com recursos limitados como a ESP32-CAM.

5. **Conversão para String Base64:** Para facilitar o envio dos dados, o array de bytes da imagem é convertido para uma string no formato base64. O processo de conversão da imagem capturada pela ESP32-CAM para uma string no formato base64 é realizado através de uma função implementada especificamente para essa tarefa, com o auxílio da biblioteca Base64.h. O resultado dessa função de conversão é uma string de caracteres que pode ser transmitida por meio do protocolo MQTT.

6. **Separação em Pacotes de 128 Bytes e Envio via MQTT:** Devido à limitação de transmissão da ESP32-CAM, que permite o envio de até 128 bytes por vez, a string base64 resultante é dividida em pacotes de 128 bytes. Esses pacotes são enviados sequencialmente ao módulo services utilizando o protocolo MQTT. Para sinalizar o início e o fim do envio, o dispositivo envia primeiro a string de identificação ";;START;;", seguida de todos os pacotes de dados de forma ordenada. Após o envio de todos os pacotes, a string ";;END;;"é transmitida para indicar o término do processo. Esse método garante que o módulo services receba e reconstrua o arquivo de imagem na sequência correta após o recebimento de todos os pacotes.

3.3 Módulo Banco de dados

1. **Modelagem:**

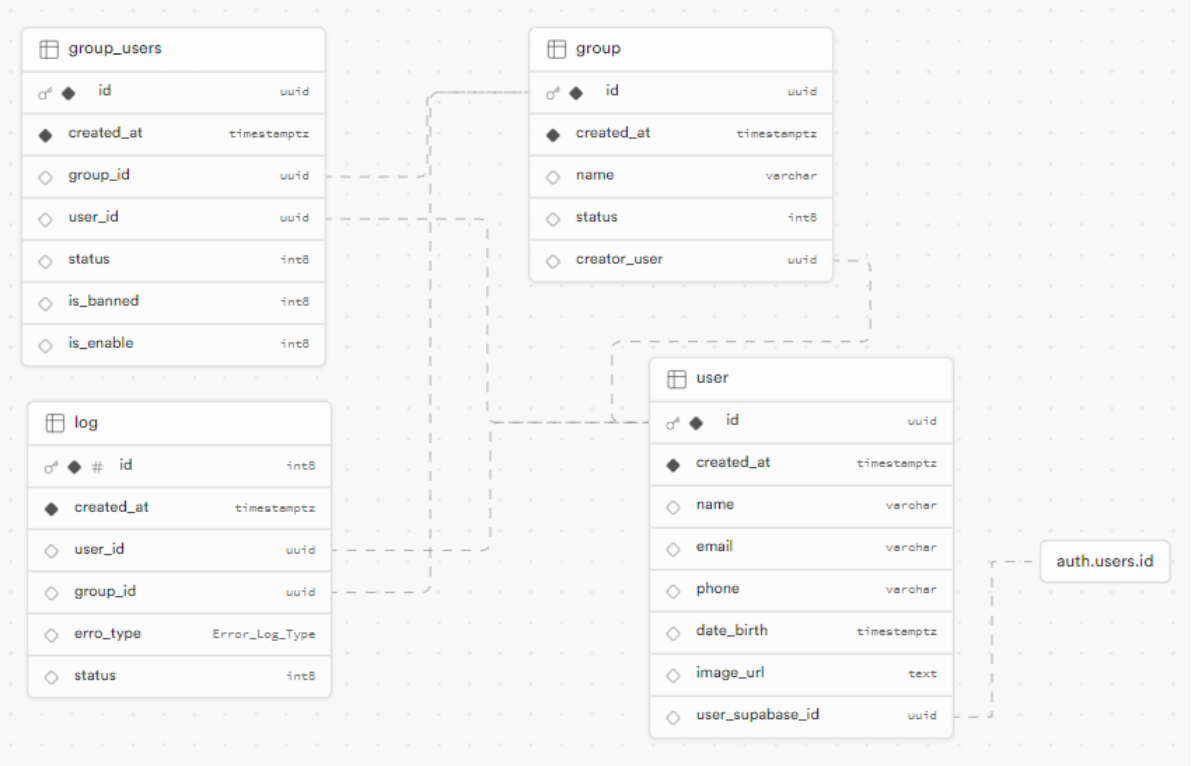

Figura 3: Modelagem das tabelas

Tabela 1: Tabela X Funções

Tabela	Função
users	Armazena as informações referentes aos usuários
group	Armazena as informações referentes aos grupos/organizações
group_users	Estabelece a relação entre os administradores e os grupos
log	Registra informações sobre o processo de verificação facial dos clientes

Como mostrado na Figura 3 e resumido na Tabela 1, o modelo de banco de dados foi estruturado em quatro tabelas principais, cada uma com um papel específico no gerenciamento dos dados da aplicação. A tabela "users"é responsável por armazenar as informações dos usuários, como nome, e-mail, número de telefone, senha, data de nascimento e foto de perfil, além de armazenar o id de ligação com a tabela de usuários do supabase "auth.users". Essas informações são fundamentais para a autenticação e o gerenciamento de cada usuário no sistema. A tabela "groups"tem o objetivo de permitir que os administradores classifiquem seus clientes em diferentes grupos. Nessa tabela, são armazenados dados como o nome do grupo, a data de criação e o administrador associado, facilitando o gerenciamento e a organização de clientes por categorias definidas pelo administrador. A tabela "user_group"estabelece a relação entre os administradores e os grupos, representando os clientes que estão associados a um determinado administrador e a um grupo específico. No contexto do sistema, o termo "cliente"refere-se a um usuário que foi registrado no sistema por um administrador autenticado. Por fim, a tabela "log"é responsável por registrar informações sobre o processo de verificação facial dos clientes, incluindo o sucesso ou falha da verificação e os tipos de erro encontrados durante o processo. Esses logs são fundamentais para o acompanhamento e monitoramento do desempenho do sistema de autenticação facial.

2. **Verificações de segurança:** Para reforçar a segurança no acesso e manipulação dos dados, foram implementadas políticas de Row-Level Security (RLS) no PostgreSQL para todas as tabelas. Essas políticas garantem que apenas usuários autenticados possam realizar operações nas tabelas. Assim, todas as tabelas "users", "groups", "user_group"e "log"permitem operações somente a usuários que tenham passado pelo processo de autenticação, assegurando que a manipulação dos dados esteja restrita a usuários validados pelo sistema.

3.4 Módulo Storage

1. **Modelagem:** O modelo de armazenamento de arquivos foi organizado em dois buckets. O bucket "users"é utilizado para armazenar as imagens de perfil dos usuários, com cada imagem sendo nomeada com o ID do usuário correspondente. O bucket "images"é dedicado ao armazenamento das imagens dos clientes usadas no processo de verificação facial, com cada imagem sendo nomeada com a data e hora atuais. Essa estrutura de armazenamento em buckets facilita a organização e recuperação dos dados de imagem, contribuindo para a eficiência do sistema.

2. **Verificações de segurança:** Os buckets de armazenamento também estão protegidos por políticas de segurança Row-Level Security (RLS), garantindo que apenas usuários autenticados possam realizar operações. Essas políticas são aplicadas aos buckets que armazenam imagens, correspondentes às tabelas "buckets","objects", "migrations", "s3_multipart_uploads", e "s3_multipart_uploads_parts"representadas na Figura 4. As tabelas estão localizadas dentro do schema "storage", que é um schema criado pelo Supabase para controlar e acessar as informações de arquivos criados. Para cada operação relacionada às imagens, as políticas de RLS asseguram que apenas usuários autenticados tenham acesso. Isso garante que o controle sobre as imagens seja restrito e seguro. Sempre que uma nova imagem é criada, ela é adicionada ao schema "storage"dentro do Supabase e modifica as tabelas garantindo que a estrutura de armazenamento e a segurança estejam alinhadas com as práticas recomendadas para proteção de dados.

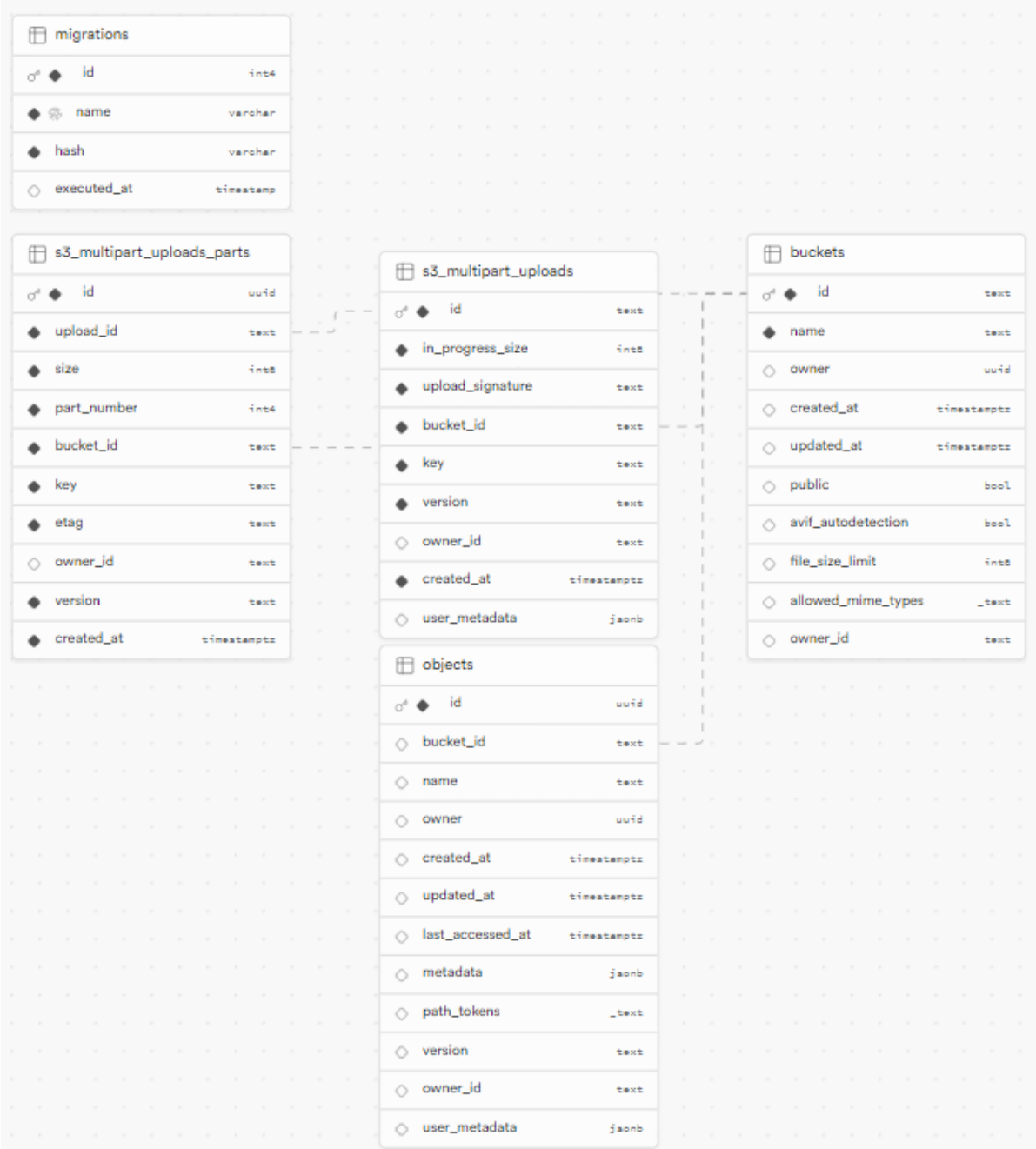

Figura 4: Modelagem das tabelas - Storage

3.5 Módulo Services

O módulo services é o componente principal do sistema, responsável por fornecer todas as funcionalidades de controle e realizar a integração entre os diferentes módulos. Para o projeto, foram utilizadas apenas APIs internas, ou seja, APIs que podem ser acessadas apenas por serviços dentro do escopo da aplicação, não permitindo chamadas de serviços externos. Na inicialização da aplicação, é instanciado um cliente do Supabase, que proverá a maior parte das funcionalidades utilizadas, conforme descrito a seguir.

1. **Autenticação:** A autenticação do sistema é gerenciada pelo Supabase, que utiliza

tokens JWT para fornecer um sistema seguro de login e logout. Embora o Supabase ofereça funcionalidades como cadastro, redefinição de senha e verificação de e-mail, no contexto da aplicação desenvolvida foi optado por utilizar apenas os recursos de registro, login e logout, com o objetivo de simplificar o desenvolvimento. O controle de acesso é garantido por usuários autenticados, e as sessões e tokens de acesso são gerenciados por meio de chamadas à API do Supabase.

2. **Funcionalidades principais:**

   (a) **Operações de CRUD no banco de dados:** As operações de criação, leitura, atualização e exclusão (CRUD) no banco de dados são realizadas utilizando as funções providas pelo Supabase. Cada operação é tratada com o devido tratamento de erros, garantindo que falhas no processo, como falha de autenticação ou dados incorretos, sejam devidamente capturadas e manipuladas. O Supabase permite interações eficientes com o PostgreSQL subjacente, integrando as funcionalidades do banco de dados diretamente na aplicação.

   (b) **Operações de CRUD no storage:** Além das operações de upload e download providas pelo Supabase, o gerenciamento dos buckets é fundamental para a organização dos arquivos no sistema. As operações de criação, leitura, atualização e exclusão de arquivos nos buckets são realizadas através das APIs do Supabase, que também possibilitam a criação e organização de diretórios dentro dos buckets. Para garantir uma operação segura, cada acesso aos buckets é feito por usuários autenticados, seguindo as políticas de segurança do sistema. As operações de gerenciamento, como criar e excluir arquivos ou pastas, são tratadas com a devida manipulação de erros, garantindo o correto funcionamento e a integridade dos dados armazenados.

   (c) **Gerenciamento do MQTT:** Foi implementada uma classe no padrão singleton utilizando a biblioteca mqtt.js. O padrão singleton define uma classe que garante que apenas uma instância dessa classe será criada e usada durante todo o ciclo de vida da aplicação, centralizando o gerenciamento de um recurso específico. Essa classe foi configurada com as mesmas definições de broker usadas pela ESP32-CAM, incluindo funcionalidades como inscrever-se em tópicos, cancelar inscrições, publicar mensagens, iniciar e encerrar conexões. Um objeto é instanciado a partir dessa classe, funcionando como o cliente central de MQTT da aplicação, responsável por gerenciar toda a comunicação via protocolo MQTT.

   (d) **Integração com outros módulos:** A integração entre os módulos é feita utilizando APIs internas do framework Next.js. Essa estratégia facilita a comunicação entre os diferentes módulos do sistema, permitindo que as funcionalida-

des sejam acessadas de forma estruturada e segura, mantendo a modularidade e garantindo que cada parte do sistema interaja de maneira eficiente.

3.6 Módulo IA

O módulo de Inteligência Artificial (IA) tem como principal função processar as imagens recebidas e realizar a identificação e comparação facial, atuando diretamente no reconhecimento facial para autenticar os usuários da aplicação.

1. **Configuração:** Na fase de configuração do módulo de IA, os modelos utilizados pelo face-api.js são carregados durante a inicialização do sistema para otimizar o tempo de processamento. O face-api.js, uma biblioteca JavaScript baseada em TensorFlow.js, foi escolhida pela sua capacidade de realizar reconhecimento facial diretamente no ambiente do navegador ou backend, eliminando a necessidade de servidores externos e reforçando a privacidade dos dados. Essa biblioteca inclui modelos pré-treinados otimizados para detecção e reconhecimento facial em tempo real, facilitando a integração em sistemas de IoT como o implementado neste projeto.

   Durante a configuração, são carregados modelos específicos: o ssd_mobilenetv1_model, que realiza uma detecção de faces na imagem capturada com precisão e eficiência; o tiny_face_detector_model, que é uma versão mais leve para detecção rápida em dispositivos com recursos limitados; o face_landmark_68_model, que identifica 68 pontos faciais essenciais, como os contornos dos olhos, boca e mandíbula; e o FaceRecognizerNet, que é responsável pela comparação e reconhecimento facial, associando as faces detectadas com aquelas armazenadas no banco de dados. Esse pré-carregamento permite que o módulo IA inicie o processamento de imagens de forma imediata, minimizando a latência e maximizando a eficiência.

2. **Funcionamento do Modelo:** Ao ser acionado, o módulo de IA recebe a imagem capturada pela ESP32-CAM e aplica os modelos carregados para extrair as características faciais da imagem. Inicialmente, a face é detectada e os pontos faciais são mapeados, possibilitando a identificação de informações adicionais, como idade e gênero. Esses dados auxiliam no refinamento da busca no banco de dados, otimizando a comparação e priorizando potenciais correspondências com base nas estimativas de idade e gênero.

   Após a filtragem inicial, o modelo FaceRecognizerNet utiliza os pontos faciais (facial landmarks) extraídos para realizar a comparação. A distância euclidiana entre os pontos faciais mapeados na imagem capturada e os armazenados no banco de dados é calculada, permitindo determinar o grau de similaridade entre as faces. Se a comparação atingir o nível de similaridade exigido para a autenticação, os dados do

usuário são enviados ao módulo de serviços, que conclui o processo de autenticação e registra a ação no sistema. Caso a validação não seja bem-sucedida, a tentativa de autenticação e a falha são registradas para futuras consultas.

3. **Execução do Modelo no Backend:** O processamento do reconhecimento facial é executado no backend, e não diretamente na ESP32-CAM, por várias razões. Primeiramente, o objetivo do sistema é centralizar a gestão de autenticação, e realizar essa tarefa no backend permite manter todos os registros de autenticação e configurações de segurança de forma consolidada, facilitando o monitoramento e a manutenção do sistema. Além disso, a ESP32-CAM possui recursos de processamento e memória limitados; a execução de algoritmos complexos de reconhecimento facial neste dispositivo poderia não apenas aumentar seu consumo de energia, mas também elevar sua temperatura, reduzindo potencialmente a vida útil do microcontrolador.

   Outra vantagem de utilizar o backend é a possibilidade de escalar o sistema de forma mais eficiente. No backend, é possível utilizar recursos de processamento robustos, adaptáveis ao volume de dados e número de usuários, permitindo que múltiplas requisições de autenticação sejam tratadas simultaneamente. O backend também permite atualizações e melhorias dos modelos de IA sem a necessidade de reconfigurar ou substituir as unidades ESP32-CAM em campo. Essa abordagem garante um sistema mais seguro, escalável e de fácil manutenção, além de minimizar os riscos de falhas no hardware da ESP32-CAM decorrentes do processamento intenso.

3.7 Ferramentas e Tecnologias

1. **Supabase:** O Supabase é uma plataforma de backend como serviço (BaaS) que oferece um conjunto de ferramentas para desenvolvimento de aplicativos com base em tecnologias abertas, como o PostgreSQL. Ele simplifica o desenvolvimento ao fornecer uma API pronta para ser consumida, além de recursos como autenticação de usuários, armazenamento de arquivos, e funcionalidades em tempo real. Ao utilizar o Supabase, desenvolvedores podem integrar rapidamente funcionalidades essenciais, sem a necessidade de configurar ou gerenciar servidores manualmente. A arquitetura baseada em PostgreSQL permite consultas complexas e escalabilidade, o que faz do Supabase uma alternativa robusta a outras plataformas de BaaS, como Firebase, mas com o diferencial de ser uma solução de código aberto. No projeto, o Supabase foi utilizado para simplificar o desenvolvimento do sistema web, com a integração de funcionalidades críticas, como autenticação, API, banco de dados, armazenamento e funcionalidades em tempo real. Ao fornecer uma solução pronta para uso, o Supabase reduziu significativamente o tempo de desenvolvimento e pos-

sibilitou o foco nas funcionalidades principais do sistema, como o gerenciamento de autenticação facial e a manipulação dos dados transmitidos.

2. **Next:** O Next.js é um framework de desenvolvimento front-end baseado em React, que oferece um ambiente otimizado para o desenvolvimento de aplicações web modernas, combinando renderização no servidor (SSR) e geração de páginas estáticas (SSG). Ele facilita a criação de aplicações web dinâmicas e interativas, com otimizações automáticas de desempenho, como divisão de código e pré-carregamento de páginas. Além disso, o Next.js possui uma forte integração com APIs, permitindo a comunicação eficiente com backends e serviços externos. Sua flexibilidade na manipulação do estado da aplicação e suporte a rotas dinâmicas faz dele uma escolha ideal para projetos que demandam interfaces ricas e interativas. Neste projeto, o Next.js foi escolhido por sua simplicidade e capacidade de acelerar o desenvolvimento, permitindo uma manipulação eficiente do estado da aplicação no lado do cliente. Sua integração com as APIs fornecidas pelo Supabase e sua arquitetura modular tornaram o processo de desenvolvimento ágil, mantendo a qualidade e a escalabilidade do sistema.

3. **MQTT:** O MQTT (Message Queuing Telemetry Transport) é um protocolo de comunicação leve e eficiente, ideal para dispositivos conectados em redes com largura de banda limitada ou instáveis. Ele opera sobre o modelo publish/subscribe, onde dispositivos podem publicar mensagens em tópicos e outros podem se inscrever para receber essas mensagens. O MQTT foi projetado para ser extremamente eficiente em termos de consumo de energia e largura de banda, tornando-o ideal para aplicações IoT (Internet das Coisas) e M2M (Machine-to-Machine). Devido à sua arquitetura, o protocolo também oferece baixo consumo de recursos, o que o torna adequado para dispositivos embarcados com recursos limitados. No contexto deste projeto, o MQTT foi empregado para garantir a comunicação eficiente entre a ESP32-CAM e o backend. No lado do servidor, o pacote mqtt.js foi utilizado para gerenciar essa comunicação, enquanto na ESP32-CAM, as bibliotecas WiFi.h e PubSubClient.h permitiram a conexão da câmera ao broker MQTT e o envio das imagens capturadas. A escolha do protocolo se deu devido à sua leveza e capacidade de funcionar de forma eficaz em redes limitadas, garantindo uma comunicação confiável.

4. **Faceapi.js:** O face-api.js é uma biblioteca JavaScript de código aberto construída sobre o TensorFlow.js, desenvolvida por Vincent Mühler. Seu propósito principal é fornecer uma série de APIs de alto nível para detecção, reconhecimento e análise facial em aplicações web ou em servidores. Ao ser implementada inteiramente em JavaScript, a biblioteca oferece a vantagem de ser executada diretamente no navegador ou no servidor, sem a necessidade de recorrer a serviços de terceiros ou depender de plataformas externas para o processamento facial. A arquitetura

do face-api.js é baseada em redes neurais profundas que permitem a extração de características faciais (landmarks) diretamente das imagens. Essas características incluem pontos-chave do rosto, como a posição dos olhos, nariz, boca, entre outros, que são utilizados para calcular uma "assinatura facial"única para cada rosto. No processo de autenticação, a biblioteca não utiliza um modelo treinado especificamente com as imagens dos usuários, mas sim um método de comparação entre essas assinaturas, também conhecidas como embeddings faciais. O modelo extraído pelo face-api.js não é treinado diretamente com os rostos dos usuários, mas faz a comparação entre duas imagens por meio da distância euclidiana entre os embeddings faciais, ou seja, entre as representações vetoriais das características faciais extraídas de cada imagem. Quando a distância euclidiana entre essas características é suficientemente pequena, o sistema reconhece que as duas imagens pertencem ao mesmo indivíduo. Esse processo garante que o sistema de autenticação facial seja preciso sem a necessidade de treinar um modelo com os dados dos usuários diretamente, proporcionando um nível adicional de segurança e privacidade. A biblioteca oferece suporte a várias funcionalidades, como detecção de múltiplos rostos em uma única imagem, detecção de pontos de referência (landmarks) no rosto, detecção de emoções faciais e verificação de identidade. O uso de face-api.js no projeto garantiu uma implementação eficiente e confiável do reconhecimento facial, permitindo a autenticação dos usuários de maneira ágil, sem comprometer a precisão ou a integridade dos dados. Além disso, sua integração com outras ferramentas, como o Next.js, facilitou o desenvolvimento da aplicação como um todo, mantendo a flexibilidade no uso de tecnologia de reconhecimento facial em diferentes ambientes.

5. **ESP32-CAM (OV2640):** A ESP32-CAM é um microcontrolador desenvolvido pela Espressif, que combina um processador ESP32 com uma câmera OV2640, permitindo o desenvolvimento de aplicações de captura de imagem e vídeo com conectividade Wi-Fi. A ESP32 é conhecida por seu baixo custo e alto desempenho em redes IoT, sendo amplamente utilizada em projetos que requerem conectividade com a internet e processamento embarcado. Além de capturar imagens e vídeos, a ESP32-CAM possui recursos de transmissão de dados, o que a torna ideal para aplicações como vigilância, reconhecimento facial e captura remota de imagens. Neste projeto, a ESP32-CAM foi utilizada para capturar imagens faciais e transmiti-las para o backend por meio do protocolo MQTT. A capacidade da ESP32 de se conectar a redes Wi-Fi e transmitir dados em tempo real a tornou a escolha perfeita para o cenário de autenticação facial proposto. Além disso, sua compatibilidade com bibliotecas como WiFi.h e PubSubClient.h facilitou a integração com o protocolo MQTT, garantindo a comunicação entre o dispositivo e o servidor de forma eficiente.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção, são apresentados os resultados obtidos buscando analizar, tanto a interface final do sistema e o propósito de cada tela, como a execução dos testes de precisão e desempenho do sistema de reconhecimento facial, baseando-se nas variações de distância e iluminação. Adicionalmente, é realizada uma análise do custo total do projeto, considerando os principais componentes utilizados. Os resultados apresentados serão também comparados com o estado da arte, permitindo uma avaliação mais abrangente da eficácia e viabilidade do sistema em relação a soluções existentes no mercado.

4.1 Módulo Frontend - Resultados

O módulo frontend desempenha um papel crucial no fornecimento de acesso às funcionalidades do sistema para os usuários. Ele é responsável por apresentar a interface gráfica com a qual os usuários interagem, garantindo uma experiência de uso clara e eficiente. A seguir, são detalhados os principais aspectos de seu funcionamento:

1. **Funcionamento da autenticação:** O processo de autenticação no sistema ocorre por meio de duas telas distintas: uma dedicada ao Login 5 e outra ao Cadastro 6 de novos usuários. Ambas as interfaces interagem com o módulo de serviços, o qual se comunica com o Supabase para gerenciar as operações de autenticação. O sistema oferece uma abordagem integrada que facilita o acesso ao controle de usuários registrados, assegurando que as credenciais sejam geridas de maneira segura.

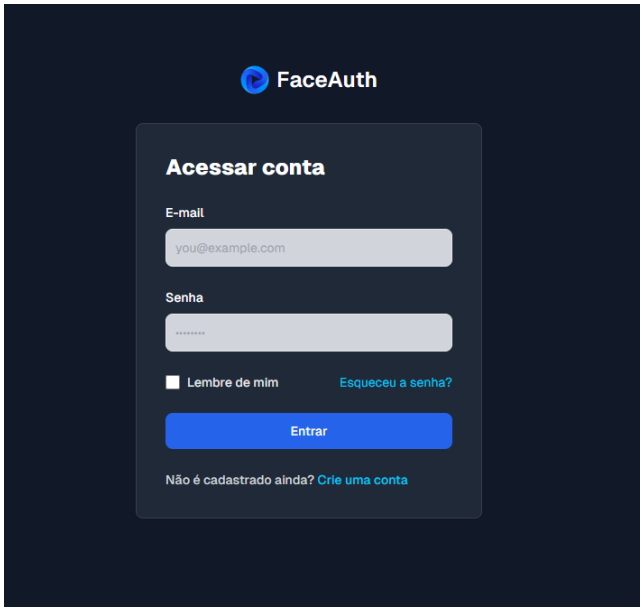

Figura 5: Login

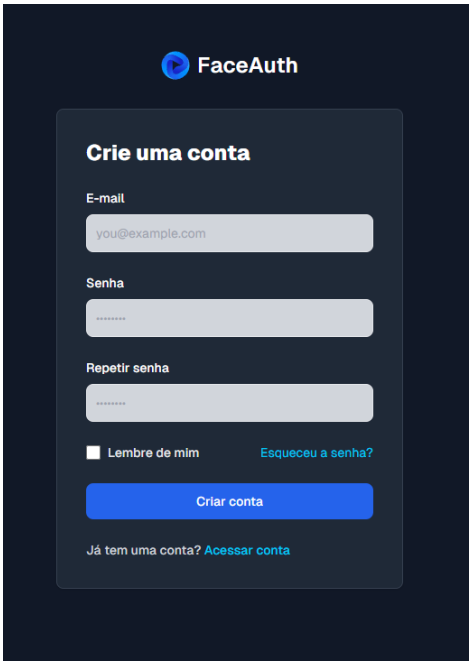

Figura 6: Cadastro

2. **Dashboard (Painel administrativo):** A tela do Painel Administrativo 7 é projetada para fornecer aos administradores o controle total sobre os usuários cadastrados no sistema. Através dela, é possível realizar operações como banimento, exclusão de contas e habilitação ou desabilitação da autenticação de usuários. Além disso, o painel apresenta gráficos que exibem um resumo dos logs de autenticação dos últimos 30 dias, permitindo uma visualização rápida e eficaz de informações relevantes para a gestão de acessos.

Figura 7: Dashboard

3. **Cadastro de novos usuários:** A tela de Cadastro de Novos Usuários 8 tem como principal objetivo facilitar a inserção de clientes no sistema de autenticação. Nessa interface, o administrador de um grupo pode registrar novos usuários por meio de um formulário que, além dos dados tradicionais, permite a adição de uma imagem. Esta imagem pode ser capturada diretamente pela ESP32-CAM, ou enviada através de upload, ampliando as possibilidades de uso e integração.

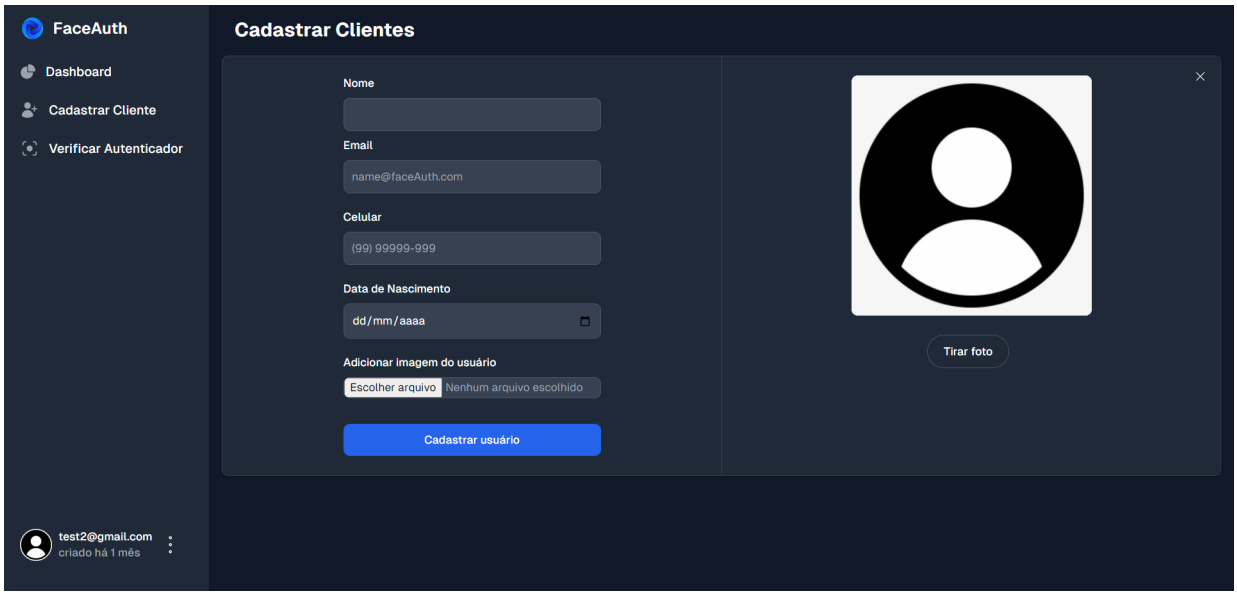

Figura 8: Cadastro de Novos Usuários

4. **Verificação de autenticação facial:** A tela de Verificação de Autenticação Facial 9 foi projetada com o propósito de demonstrar o funcionamento prático do sistema de autenticação. Sua implementação simplificada permite que, ao acionar o botão de autenticação, o módulo de serviços execute todo o ciclo de tratamento necessário para validar ou invalidar a tentativa de acesso. Esse fluxo integra diferentes etapas de processamento e garante a segurança e integridade do sistema.

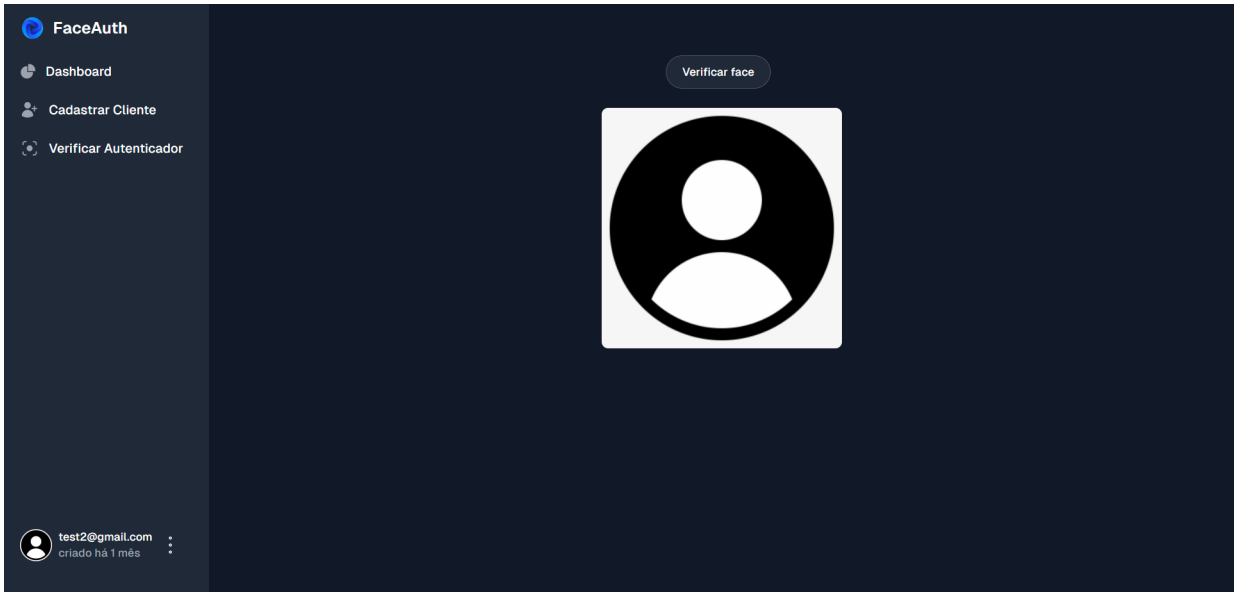

Figura 9: Verificação de Autenticação Facial

5. **Outras telas (perfil, 404, marketing):** Além das telas descritas anteriormente, o sistema conta com outras interfaces que complementam suas funcionalidades. A tela de Perfil 10 permite que o usuário personalize sua conta, ajustando configurações e dados pessoais. A tela de Erro 11, ou "404", é apresentada sempre que uma rota inválida é acessada ou quando ocorre um erro inesperado, garantindo uma resposta clara ao usuário. Por fim, a tela de Marketing 12 serve como ponto de entrada do sistema para usuários não autenticados, sendo utilizada para divulgar informações e atrair novos usuários.

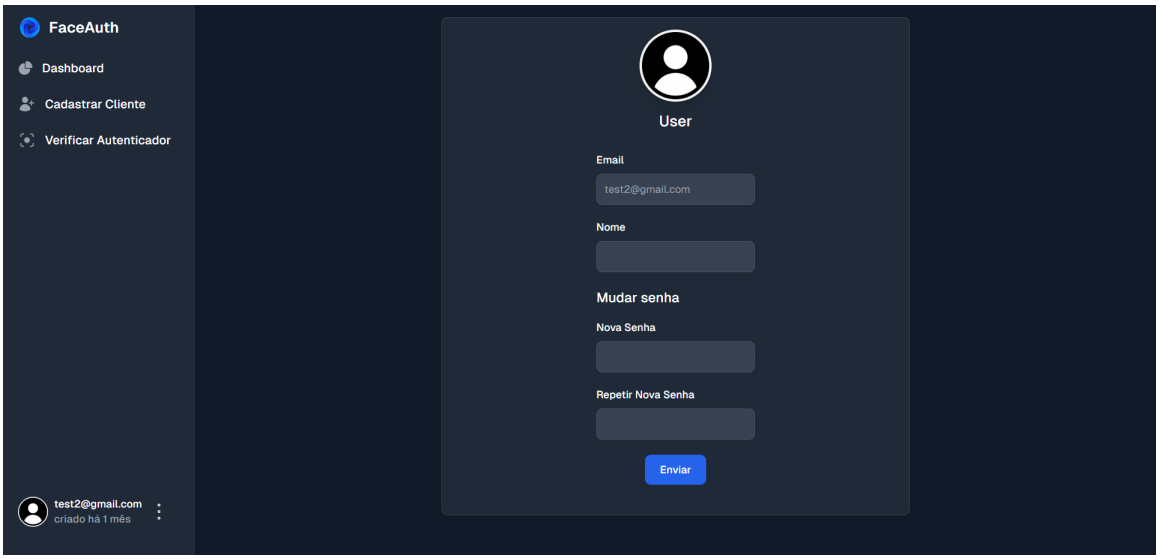

Figura 10: Perfil

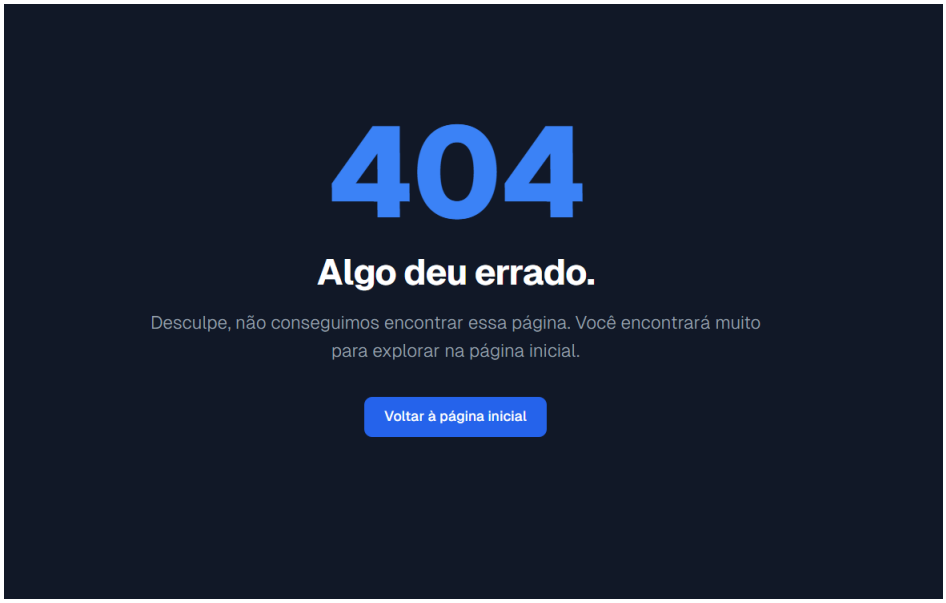

Figura 11: Erro - 404

Figura 12: Marketing

4.2 Procedimentos de Teste

Os testes descritos nesta seção foram conduzidos para avaliar a precisão (acurácia) do sistema de reconhecimento facial em diferentes condições. Foram realizados testes com variação de distância e iluminação, buscando entender como essas variáveis influenciam o desempenho do sistema. Para os testes, foram utilizados dois indivíduos, ambos do sexo masculino, de cor parda, com idades de 22 e 24 anos. Em cada teste, foram considerados a taxa de acertos e a eficácia geral do sistema em reconhecer corretamente os indivíduos.

1. **Teste de Precisão (Acurácia)**
   O teste de precisão teve como objetivo verificar a capacidade do sistema de au-

tenticação facial de identificar corretamente os indivíduos em diferentes distâncias. Neste experimento, quatro distâncias foram utilizadas: 30 cm, 60 cm, 1 metro e 1 metro e 30 cm. Em cada distância, foram realizados 20 testes, sendo 10 para o indivíduo de 22 anos e 10 para o de 24 anos. Todos os testes ocorreram sob uma condição de iluminação média.

(a) **Testes com variação de distância**

Para medir a precisão do sistema em distâncias diferentes, os indivíduos foram posicionados a 30 cm, 60 cm, 1 metro e 1 metro e 30 cm da câmera. A iluminação média foi definida como um nível de luz artificial moderada, como a presente em um ambiente interno bem iluminado, sem fontes diretas de luz intensa. Os resultados obtidos para cada distância estão representados na Tabela 2, com a taxa de acertos em relação ao total de testes realizados.

Tabela 2: Acurácia - Distância

Distância	**Acertos/Total**	**Acurácia (%)**
30 cm	19/20	95%
60 cm	19/20	95%
1 m	18/20	90%
1 m 30 cm	17/20	85%

A acurácia média final foi obtida somando-se as taxas de acerto de cada distância e dividindo o resultado pelo número de distâncias testadas, como no cálculo abaixo:

$$\text{Acurácia Média} = \frac{95\% + 95\% + 90\% + 85\%}{4} = 91.25\%$$

(b) **Testes com variações de iluminação**

O segundo conjunto de testes foi realizado para avaliar o impacto de diferentes condições de iluminação no desempenho do sistema de reconhecimento facial. Para esses testes, a distância foi mantida fixa em 60 cm, que apresentou os melhores resultados nos testes de variação de distância. Foram simuladas três condições de iluminação: pouca, média e muita luz.

Definição das condições de iluminação:

i. **Pouca iluminação:** Refere-se a um ambiente com luz fraca ou indireta, como em uma sala com iluminação artificial limitada ou luz natural indireta.

ii. **Média iluminação:** Corresponde a um nível de luz adequado para uma boa captura de imagens, como um ambiente interno bem iluminado por luz artificial distribuída uniformemente.

iii. **Muita iluminação:** Refere-se a ambientes com luz intensa, como sob luz solar direta ou com luzes artificiais fortes direcionadas.

Os resultados obtidos em cada nível de iluminação estão detalhados na Tabela 3.

Tabela 3: Acurácia - Iluminação

Iluminação	**Acertos/Total**	**Acurácia (%)**
Pouca	17/20	85%
Média	19/20	95%
Muita	19/20	95%

A acurácia média final considerando as variações de iluminação foi calculada como:

$$\text{Acurácia Média} = \frac{85\% + 95\% + 95\%}{3} = 91.67\%$$

Para calcular a acurácia média final do sistema, considerando tanto os testes de variação de distância quanto os testes de variação de iluminação, foi feita a média entre os dois valores obtidos nas subseções anteriores:

$$\text{Acurácia Média Final} = \frac{91.25\% + 91.67\%}{2} = 91.46\%$$

Esse valor representa a taxa de acerto média do sistema de reconhecimento facial, considerando as diferentes distâncias e condições de iluminação testadas.

2. **Teste de Desempenho (Média do tempo de resposta do sistema):**

   (a) **Teste com usuários cadastrados**

   O teste de desempenho para usuários cadastrados foi realizado com o objetivo de medir o tempo de resposta do sistema desde o momento em que o usuário clica no botão de autenticação até o recebimento da validação da identidade pelo sistema. Para isso, foram utilizados os métodos da função Date do JavaScript, que capturam o tempo exato de início e fim do processo de autenticação. Os testes foram realizados com 50 tentativas de autenticação de usuários previamente cadastrados no sistema. O tempo médio de resposta registrado para usuários cadastrados foi de 7,31 segundos. Esse valor foi calculado a partir da média de todos os tempos registrados durante as 50 tentativas. O sistema demonstrou uma performance consistente, com pequenas variações dependendo da latência da rede e das condições do servidor durante os testes.

(b) **Teste com usuários não cadastrados**

Para usuários não cadastrados, o procedimento de teste foi semelhante ao realizado com usuários cadastrados. Foram realizadas 50 tentativas de autenticação, sendo que o sistema, ao identificar que o usuário não estava registrado, rejeitava a tentativa de login. Assim como no teste anterior, os métodos da função Date do JavaScript foram usados para medir o tempo de resposta entre o clique do usuário no botão de autenticação e o retorno da resposta do sistema. O tempo médio de resposta para usuários não cadastrados foi de 10,4 segundos. Esse tempo foi maior em comparação com os usuários cadastrados devido ao processo de verificação no banco de dados, que necessitava confirmar que o usuário não estava presente no sistema, aumentando o tempo necessário para o retorno da resposta.

4.3 Gráficos - Acurácia

1. **Acurácia - Distância**

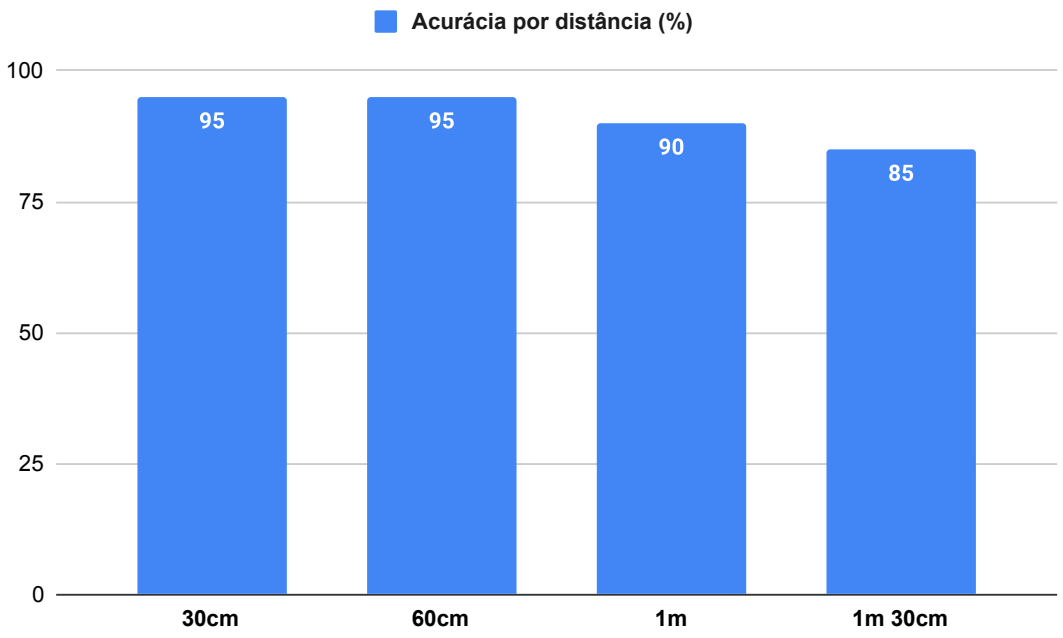

Figura 13: Acurácia - Distância

Os resultados apresentados na Figura 13 demonstram que o sistema de reconhecimento facial apresentou uma acurácia consistente de 95% nas distâncias de 30 cm e 60 cm. Isso indica que, em distâncias curtas, o sistema é capaz de identificar com precisão os indivíduos, possivelmente devido à maior resolução das imagens capturadas pela câmera, que facilita o processamento dos traços faciais.

No entanto, ao aumentar a distância para 1 metro, observa-se uma redução da acurácia para 90%, conforme observado na Figura 13. Esse decréscimo pode ser atribuído à diminuição da qualidade da imagem capturada, uma vez que, quanto maior a distância, menores e menos definidos se tornam os detalhes faciais nas imagens processadas.

A uma distância de 1 metro e 30 cm, a acurácia diminuiu ainda mais, atingindo 85%. Esse resultado sugere que o sistema encontra dificuldades para manter a precisão em distâncias maiores, uma vez que a qualidade das imagens continua a se deteriorar à medida que o indivíduo se afasta da câmera.

A acurácia média, considerando todas as distâncias testadas, foi de 91,25%. Esse valor reflete o desempenho geral do sistema em diferentes cenários de distância e indica que, apesar da alta acurácia em distâncias curtas, a eficácia diminui gradualmente conforme o indivíduo se afasta da câmera. Esses dados evidenciam a importância de posicionar a câmera a uma distância adequada para maximizar a precisão do reconhecimento facial.

2. **Acurácia - Iluminação**

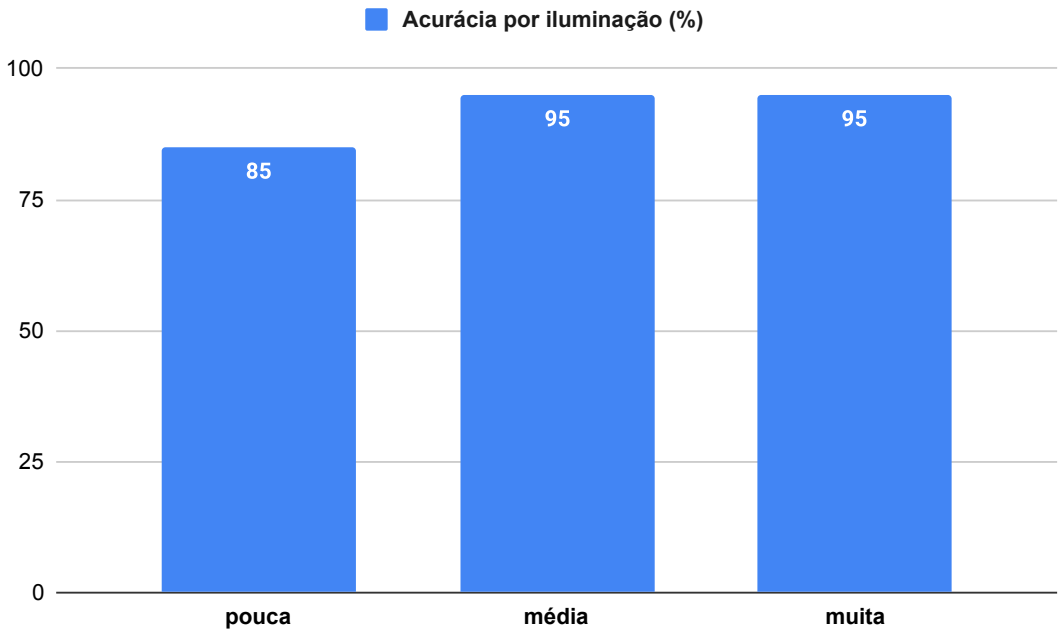

Figura 14: Acurácia - Iluminação

Os resultados ilustrados na Figura 14 indicam que o desempenho do sistema de reconhecimento facial é diretamente influenciado pelas condições de iluminação. Sob condições de média e muita iluminação, o sistema atingiu sua máxima eficiência, com uma acurácia de 95%. Isso demonstra que, em ambientes com iluminação

adequada, a câmera é capaz de capturar imagens com qualidade suficiente para que o algoritmo de reconhecimento facial identifique os indivíduos de maneira confiável.

Em contraste, nas condições de pouca iluminação, a acurácia caiu para 85%, conforme observado na Figura 14. Esse resultado evidencia que a baixa luminosidade afeta negativamente a qualidade das imagens, prejudicando a definição dos traços faciais. Ambientes com pouca luz geram imagens menos nítidas, nas quais o sistema encontra maior dificuldade para distinguir características faciais com precisão.

A acurácia média obtida ao longo dos três cenários de iluminação foi de 91,67%. Esse valor reflete o comportamento geral do sistema diante de diferentes intensidades de luz. Embora o sistema apresente alta acurácia em condições de iluminação favoráveis, as limitações em ambientes com pouca luz ressaltam a importância de garantir condições adequadas de iluminação para otimizar o desempenho do reconhecimento facial.

4.4 Gráficos - Desempenho

Os resultados de desempenho indicam uma diferença significativa no tempo de resposta entre usuários cadastrados e não cadastrados no sistema, conforme ilustrado na Figura 15.

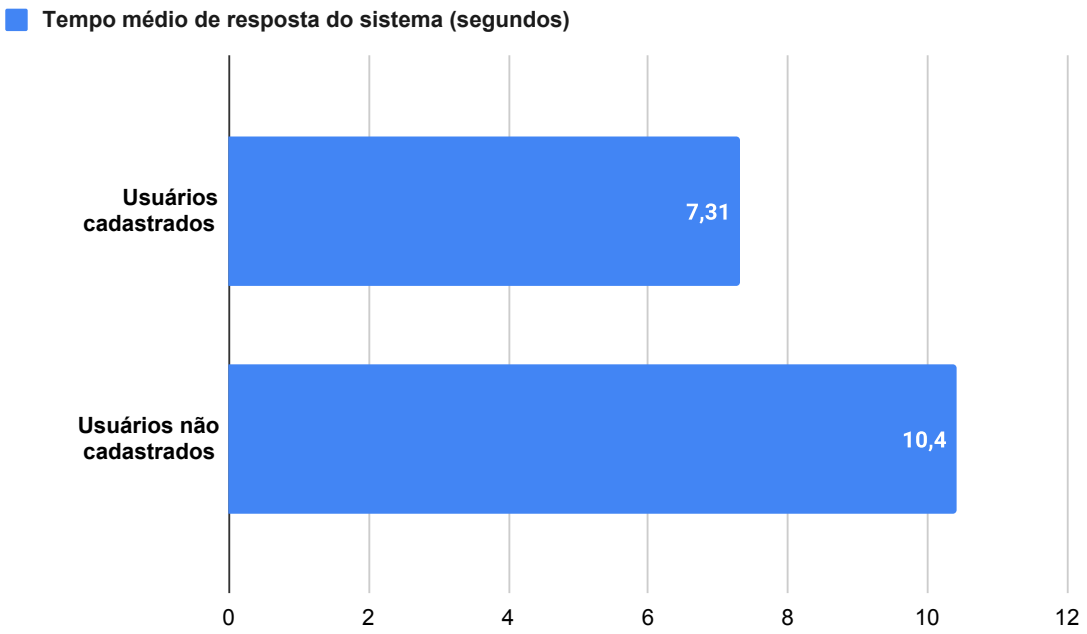

Figura 15: Tempo Médio de Resposta do Sistema

1. **Desempenho com Usuários Cadastrados**
   O tempo médio de resposta para usuários cadastrados foi de 7,31 segundos. Esse

valor reflete a eficiência do sistema ao autenticar indivíduos previamente registrados. A estabilidade do tempo de resposta indica que o sistema mantém um desempenho consistente durante o processamento de autenticações. A pequena variação nos tempos pode ser atribuída a fatores externos, como latência de rede ou carga do servidor, porém esses desvios foram mínimos e não comprometeram a eficiência do sistema.

2. **Desempenho com Usuários Não Cadastrados**
   Para usuários não cadastrados, o tempo médio de resposta foi de 10,4 segundos, conforme mostrado na Figura 15. Esse aumento no tempo, comparado aos usuários cadastrados, sugere que a verificação adicional no banco de dados prolonga o processo de resposta do sistema. A diferença de 3,09 segundos entre os dois cenários indica que o sistema demora mais ao identificar que o usuário não está registrado, aumentando o tempo necessário para a rejeição da autenticação.

4.5 Analisando o Custo do Projeto

Ao analisar os custos do projeto, observa-se que o único componente físico utilizado é a ESP32-CAM, com um preço médio de R$ 42. Esse dispositivo é responsável pela captura das imagens enviadas ao backend para o processo de reconhecimento facial, o que faz do investimento em hardware um dos principais fatores que contribuem para manter o custo inicial do projeto acessível.

No que diz respeito à infraestrutura de software, tanto o Next.js quanto o Supabase são tecnologias de código aberto, o que elimina a necessidade de gastos com licenciamento. Ambas as ferramentas podem ser implantadas localmente sem custos adicionais, desde que haja uma máquina disponível, como um computador com pelo menos 8 GB de RAM e um processador intermediário como um i3 (Processador Intel Core i3-12300HE), que atenda às dependências básicas, como Docker e Node. Essa abordagem dispensa o uso de um equipamento de alto desempenho, permitindo que o processamento de dados e a hospedagem do backend sejam realizados de forma eficiente em um computador pessoal. Esse cenário é ideal para minimizar os investimentos iniciais e evitar os custos recorrentes de hospedagem em nuvem.

Por outro lado, para quem não deseja manter um servidor localmente, existe a opção de hospedar o projeto em serviços de cloud computing, como Google Cloud, AWS, Azure ou nas próprias plataformas do Supabase e Vercel (para Next.js). O projeto foi estruturado para permitir a implantação em diversas opções de nuvem.

A implantação em cloud oferece vantagens, como alta disponibilidade, garantindo que o sistema esteja sempre online, sem a necessidade de manutenção física. Além disso, os serviços de nuvem permitem escalabilidade, ajustando os recursos conforme a demanda,

e oferecem segurança e manutenção da infraestrutura, gerenciadas pelos provedores, o que reduz a necessidade de monitoramento técnico contínuo. Contudo, a principal desvantagem dessa alternativa são os custos recorrentes, que variam de acordo com o uso de recursos como armazenamento, largura de banda e número de requisições. Dependendo da escala do projeto, esses custos podem aumentar significativamente ao longo do tempo. Além disso, a hospedagem em nuvem exige uma conexão de internet estável para o funcionamento contínuo da aplicação e o gerenciamento remoto dos recursos.

Dessa forma, pode-se concluir que a meta de desenvolvimento de um protótipo de baixo custo foi atingida. Na opção de implantação local, o único investimento inicial seria a compra da ESP32-CAM, com um custo estimado de R$ 42, somado a eventuais acessórios, como cabos e fontes de alimentação, que somariam aproximadamente R$ 46 no total. Essa solução oferece uma alternativa acessível para a construção de um sistema de reconhecimento facial, com a vantagem de ser facilmente expandido ou adaptado conforme as necessidades do usuário. Na opção de implantação em nuvem, os custos podem variar de acordo com os serviços contratados, mas a flexibilidade e os recursos oferecidos por essa modalidade tornam essa alternativa viável para projetos que exigem maior escalabilidade e disponibilidade.

4.6 Comparação com Estado da arte

Os resultados obtidos no projeto de autenticação facial utilizando a ESP32-CAM indicam uma acurácia média de 91,46%, sendo impactados por variações de distância e iluminação. Comparando com os estudos revisados, o trabalho de Yalçın et al. (2023) [1], que combina reconhecimento facial com linguagem de sinais, obteve uma acurácia ligeiramente superior, na faixa de 93%, destacando-se pela maior acessibilidade oferecida a usuários com deficiência auditiva e de fala. No entanto, enfrentaram desafios semelhantes em condições adversas de iluminação e distância, sugerindo que a ESP32-CAM apresenta limitações compartilhadas em ambos os casos.

O trabalho de Irjanto e Surantha (2020) [2], utilizando redes neurais convolucionais (CNNs), apresenta uma acurácia significativamente superior, com taxas de reconhecimento facial de até 98% em ambientes controlados, indicando que o uso de CNNs oferece uma melhora substancial na precisão. Essa abordagem, ao focar em segurança residencial, também beneficia-se de hardware mais robusto e otimizações específicas, permitindo uma detecção mais confiável de intrusos.

Já o estudo de Zhou et al. (2021) [3], que utiliza sensores acústicos junto com reconhecimento facial, foca na robustez contra ataques de spoofing. Embora o artigo não forneça diretamente uma acurácia específica para o reconhecimento facial isolado, a combinação de sensores acústicos mostrou-se altamente eficiente em evitar falsos positivos,

um problema crítico em sistemas convencionais de reconhecimento facial. A abordagem dupla (imagem e som) aprimorou significativamente a segurança, algo não abordado no projeto atual.

Por fim, o estudo de Guo et al. (2014) [4] foca em desafios relacionados ao uso de maquiagem, onde propuseram algoritmos avançados que permitiram ao sistema alcançar uma acurácia de 92% mesmo com alterações faciais causadas por maquiagem. Isso demonstra uma abordagem inovadora que resolve problemas de falsos negativos, algo que o projeto com ESP32-CAM pode enfrentar se aplicado em cenários onde a aparência facial muda frequentemente.

Concluindo, enquanto o projeto atual atinge a meta de baixo custo e uma acurácia competitiva, os trabalhos de estado da arte exploram técnicas mais robustas, como CNNs, detecção acústica e algoritmos especializados, alcançando uma maior precisão e segurança. Isso sugere que, embora o sistema desenvolvido seja funcional e acessível, existe um espaço para aprimoramentos, especialmente em cenários com alta demanda de segurança e variações extremas de condições ambientais.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi realizado o desenvolvimento de uma solução de autenticação facial eficiente, segura e de baixo custo para ambientes IoT. A primeira hipótese, que sugeria ser possível desenvolver uma solução eficiente em hardware limitado, foi confirmada com o uso da ESP32-CAM e algoritmos otimizados. Embora tenha sido observada uma redução na acurácia em distâncias superiores a 1 metro e em condições de baixa iluminação, o sistema atingiu uma média de 91,25% de precisão em diferentes condições de teste. Esse resultado é comparável aos trabalhos de estado da arte que também utilizam a ESP32-CAM, como o estudo de Yalçın et al. (2023) [1], que obteve uma acurácia similar de 93%. Conforme discutido no capítulo de resultados, a solução desenvolvida comprovou ser viável para ambientes de baixo custo.

Ao analisar a segunda hipótese, que se referia à integração segura do sistema com uma arquitetura IoT, foi possível perceber que, de fato, o protocolo MQTT ofereceu uma base confiável para essa comunicação. No entanto, uma análise mais aprofundada revela que a segurança poderia ser reforçada em vários pontos. Por exemplo, a string base64, que contém as imagens capturadas pela ESP32-CAM, poderia ser criptografada antes de ser enviada ao servidor. Isso impediria a interceptação de dados sensíveis durante a transmissão. Além disso, o armazenamento dos dados biométricos, tanto no banco de dados quanto no storage, também poderia se beneficiar de técnicas de criptografia. Atualmente, a aplicação utiliza uma comunicação criptografada com tecnologias como Next.js e Supabase, garantindo certo nível de proteção. Contudo, a ausência de criptografia nos dados em repouso (biométricos) abre brechas para vulnerabilidades. Implementar criptografia de ponta a ponta garantiria que os dados permanecessem inacessíveis a terceiros, mesmo em caso de invasão de servidores. Isso traria uma robustez significativa ao sistema, atendendo melhor à premissa de uma integração segura e elevando o nível de proteção para um padrão mais elevado, conforme esperado na hipótese inicial.

Por fim, a terceira hipótese, que sugeria um desempenho comparável ou superior a outras soluções, foi parcialmente confirmada. O tempo médio de resposta para usuários cadastrados foi de 7,31 segundos, e de 10,4 segundos para não cadastrados. Embora tenha havido variações no tempo de resposta devido à latência de rede, o sistema demonstrou consistência nos testes. Em comparação com outras soluções de alta performance, como as que utilizam CNNs (com acurácias de até 98% em ambientes controlados), o projeto atingiu resultados satisfatórios, especialmente considerando suas limitações de hardware e o foco em baixo custo.

Entre as limitações do trabalho, destacam-se a sensibilidade do sistema à variação de iluminação e a necessidade de hardware mais potente para melhorar a velocidade de resposta em soluções mais exigentes. A ESP32-CAM, embora tenha se mostrado eficaz,

limita o uso de algoritmos mais complexos devido ao seu poder de processamento limitado.

Como trabalhos futuros, seria interessante explorar a implementação de técnicas mais avançadas de reconhecimento, como redes neurais convolucionais (CNNs), que poderiam ser treinadas para otimizar a acurácia do sistema, especialmente em ambientes adversos de iluminação e com maior variação de distâncias. Outra oportunidade de expansão seria a implementação de um sistema de autenticação multifatorial, combinando a detecção facial com outras formas de autenticação, como biometria ou verificação via dispositivos móveis, para aumentar a segurança em cenários críticos. A adaptação do sistema para ser mais resiliente a ataques de spoofing ou de mudanças na aparência facial também representaria um importante avanço no estado da arte.

Conclui-se, assim, que o sistema de autenticação facial proposto cumpriu os objetivos iniciais, oferecendo uma solução acessível, de fácil implantação e com resultados comparáveis a sistemas de maior custo, corroborando com a premissa de que é possível implementar sistemas de autenticação facial em ambientes IoT com hardware restrito.

REFERÊNCIAS

[1] Yalçın, Z., Türkdağlı, O., Dalkılıç, G., Aydın, Ö. (2023). "Authentication with face recognition and sign language using ESP32-CAM," *Deu Muhendislik Fakultesi Fen ve Muhendislik*, vol. 25, pp. 481-489. doi: 10.21205/deufmd.2023257417.

[2] Irjanto, N., Surantha, N. (2020). "Home Security System with Face Recognition based on Convolutional Neural Network," *International Journal of Advanced Computer Science and Applications*, vol. 11. doi: 10.14569/IJACSA.2020.0111152.

[3] Zhou, B., Xie, Z., Zhang, Y., Lohokare, J., Gao, R., Ye, F. (2021). "Robust Human Face Authentication Leveraging Acoustic Sensing on Smartphones," *IEEE Transactions on Mobile Computing*, vol. PP, pp. 1-1. doi: 10.1109/TMC.2020.3048659.

[4] Guo, G., Wen, L., Yan, S. (2014). "Face Authentication With Makeup Changes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, pp. 814-825. doi: 10.1109/TCSVT.2013.2280076.

[5] Manullang, M., Ramdani, A., Migotuwio, N. (2020). "Design and Architecture of a Public Satisfaction Detection Camera Based on Facial Emotional Analysis," *IOP Conference Series: Earth and Environmental Science*, vol. 537, p. 012021. doi: 10.1088/1755-1315/537/1/012021.

[6] Thales Group. (2023). "Facial Recognition: An Overview of the Technology," *Thales Group*. Disponível em: https://www.thalesgroup.com/en/markets/digital-identity-and-security/biometrics/facial-recognition.

[7] Almulhim, M., Islam, N., Zaman, N. (2023). "A lightweight and secure authentication scheme for IoT based e-health applications," *International Journal of Information Security*, vol. 10, pp. 13945-13958.

[8] Babaie, S. (2020). "Biometric authentication: an efficient option for Internet of Things applications during the COVID-19 pandemic," *Acta Sci. Comput. Sci.*, vol. 2, no. 10, pp. 1-2.

[9] Módulo ESP32-CAM com Câmera OV2640 2MP - Eletrogate. Disponível em: `https://www.eletrogate.com/modulo-esp32-cam-com-camera-ov2640-2mp`. Acesso em: 8 nov. 2024.

[10] Face-api.js: JavaScript Face Recognition Leveraging TensorFlow.js. Disponível em: `https://www.infoq.com/news/2018/11/faces-api-js/`. Acesso em: 8 nov. 2024.

[11] Face API Documentation - justadudewhohacks. Disponível em: https://justadudewhohacks.github.io/face-api.js/docs/index.html. Acesso em: 8 nov. 2024.

[12] Projeto Panóptico - Centro de Estudo de Segurança e Cidadania (CESeC). Reconhecimento facial: mais de 47 milhões estão na mira das câmeras de segurança pública. Disponível em: https://revistaforum.com.br/brasil/2023/12/13/reconhecimento-facial-mais-de-47-milhes-esto-na-mira-das-cmeras-de-segurana-pu
html. Acesso em: 8 nov. 2024.

[13] Introdução ao ESP32-CAM - Eletrogate Blog. Disponível em: https://blog.eletrogate.com/introducao-ao-esp32-cam/. Acesso em: 8 nov. 2024.

[14] Acervo Lima. Arquitetura de 5 camadas da Internet das coisas. Disponível em: https://acervolima.com/arquitetura-de-5-camadas-da-internet-das-coisas/. Acesso em: 8 nov. 2024.

[15] Braga, Newton C. Comunicando-se via MQTT com o ESP32. Disponível em: https://www.newtoncbraga.com.br/index.php/microcontroladores/143-tecnologia/17117-comunicando-se-via-mqtt-com-o-esp32-mic404.html. Acesso em: 8 nov. 2024.

[16] Moreau, Luis O. ESP32-CAM MQTT Communication. Disponível em: https://github.com/luisomoreau/ESP32-cam-MQTT. Acesso em: 8 nov. 2024.