

# CLASSIC: Sistema de Chamada Escolar com Reconhecimento Facial através de IoT e Computação de Borda

Marlon G. Duarte  
Universidade Federal do Ceará  
Fortaleza, Ceará, Brazil  
marlongduarte@alu.ufc.br

Marciel B. Pereira  
Universidade Federal do Ceará  
Crateús, Ceará, Brazil  
marciel@crateus.ufc.br

Filipe de Matos  
Universidade Federal do Ceará  
Crateús, Ceará, Brazil  
filipe.fernandes@crateus.ufc.br

## RESUMO

Monitorar pessoas é uma tarefa importante para garantir a segurança de um determinado ambiente. Registrar a presença de pessoas em fábricas, escolas e outros estabelecimentos também pode servir como uma fonte valiosa de dados. Neste trabalho, empregamos tecnologias de Aprendizado de Máquina (ML) –*Machine Learning* – e Internet das Coisas (IoT) para propor o CLASSIC: um sistema inteligente para salas de aula que automatiza o controle de frequência por meio de reconhecimento facial. O sistema combina um Raspberry Pi e uma câmera para capturar imagens e recortá-las na região do rosto; esse processamento pode ser realizado com o auxílio da Computação de Borda (EC) para maior eficiência. Utilizamos um *framework* de visão computacional voltado para imagens de rostos. No *framework*, no recorte do rosto, escolhemos o *Single Shot Multibox Detector* (SSD), e para gerar as representações faciais (*face embeddings*), empregamos o ArcFace. Essa abordagem alcança uma precisão de 85% ao mesmo tempo em que reduz a latência por meio do offloading computacional. Os resultados experimentais demonstram a viabilidade do CLASSIC em cenários reais, destacando seu potencial para otimizar a gestão da sala de aula e minimizar interrupções no processo educacional.

## PALAVRAS-CHAVE

Reconhecimento Facial, IoT, *Edge Computing*, RaspberryPi, Computer Vision, Gerenciamento da Chamada Escolar

## 1 Introdução

A utilização de métodos tradicionais de lista de presença que envolvem um tipo de chamada manual para gerenciar a presença dos alunos em sala de aula consome tempo e interrompe o fluxo da aula, o que pode levar à ineficiência na gestão da turma na sala de aula. No Brasil, até 31% do tempo letivo é gasto em atividades não pedagógicas, como o controle da assiduidade, o que resulta em uma perda significativa no tempo de transmissão de conteúdos educativos [9]. Além disso, esses procedimentos podem causar desconforto aos alunos e aumentar a carga de trabalho dos educadores.

O avanço de tecnologias como Inteligência Artificial (AI) e IoT tem permitido o desenvolvimento de ferramentas inovadoras com o objetivo de automatizar processos educacionais. Particularmente, a AI baseada em Visão Computacional (CV) apresenta um potencial substancial ao permitir que os sistemas processem e interpretem imagens digitais para aplicações como o reconhecimento facial [5]. No entanto, a implementação de tais sistemas enfrenta ainda alguns desafios, incluindo elevadas exigências computacionais para o processamento de imagens e a necessidade de resposta em tempo real que excedem as capacidades dos dispositivos IoT.

Para superar essas limitações, algumas técnicas podem ser aplicadas, como o *offloading* computacional que transfere tarefas com uso intensivo de recursos para dispositivos mais potentes, como os equipamentos presentes na EC. Neste aspecto, o descarregamento do processamento para um computador de borda na rede, ao invés de usar a Computação em Nuvem (CC), aproxima o servidor da fonte de dados, reduzindo a latência e melhorando a eficiência de aplicações em tempo real, como o reconhecimento facial [4]. Esta abordagem complementa a IoT, que interconecta dispositivos físicos, permitindo a coleta e transmissão de dados para processamento inteligente e eficiente [11].

No âmbito dos ambientes educacionais, a automatização dos processos seria um caminho inicial para solucionar os problemas de desperdício de tempo de aula. Ao abordar o desafio de automatizar a frequência escolar, é fundamental reconhecer que a tecnologia de CV se apresenta como uma solução promissora, pois garante que o aluno estará dentro da sala de aula. Entretanto, é importante observar que dispositivos menos robustos computacionalmente, como aqueles inseridos no contexto de IoT, sozinhos podem não ser suficientemente poderosos para executar essa tarefa com correção e eficiência.

Este trabalho propõe o *framework Classroom Attendance System on Smart IoT and Edge Computing with Facial Recognition* (CLASSIC) para automatizar o acompanhamento da frequência de alunos. O sistema utiliza um *Raspberry Pi* equipado com uma câmera para capturar imagens dos alunos. O CLASSIC emprega uma biblioteca *Python* especializada na extração de características faciais, a qual disponibiliza um conjunto de modelos de última geração com pesos pré-treinados para a tarefa de reconhecimento facial. A integração dessa biblioteca dispensa a necessidade de treinamento de um modelo específico, otimizando o desenvolvimento da solução proposta.

O processo de reconhecimento inicia-se com a captura da imagem, seguida pela geração de um vetor de características faciais. Esse vetor é então comparado com vetores previamente armazenados em uma base de dados. A arquitetura modular e em estágios facilita a manutenção do sistema e permite a flexibilidade na seleção dos modelos de extração de características a serem utilizados.

As principais contribuições deste trabalho incluem: 1) o desenvolvimento de um sistema automatizado de controle de assiduidade destinado a reduzir a carga de trabalho dos educadores, melhorando simultaneamente a eficácia da gestão da sala de aula; 2) a avaliação do desempenho dos dispositivos IoT em tarefas de reconhecimento facial e os benefícios de descarregar tarefas computacionais para dispositivos de ponta; e 3) o reuso do *framework DeepFace*, que encapsula modelos de reconhecimento facial de última geração e executa, de forma automatizada, as etapas de detecção, alinhamento,

normalização, representação e verificação de faces, permitindo integrar ao sistema funcionalidades de análise facial com alta acurácia de forma simplificada.

Este artigo está organizado em 5 seções. Após a Introdução, a Seção 2 descreve os trabalhos relacionados e enfatiza as lacunas existentes no estado da arte. A Seção 3 é a principal do artigo, onde apresenta-se a arquitetura do sistema CLASSIC. A metodologia de avaliação do referido sistema e os resultados e discussões estão apresentados na Seção 4. Por fim, a Seção 5 apresenta as considerações finais, enfatizando como a contribuição foi atingida e tecendo trabalhos futuros para a continuidade desta pesquisa.

## 2 Trabalhos Relacionados

Nesta seção, serão apresentados os trabalhos relacionados ao tema deste estudo. A prospecção bibliográfica foi conduzida com foco em CV aplicada a contextos educacionais, particularmente na *gestão de frequência escolar*. Foram priorizadas publicações compreendidas entre 2019 e 2024 para assegurar a relevância temporal dos estudos.

A string de busca foi elaborada a partir dos termos: "face detection", "You only look once (YOLO)", "Rede Neural Profunda (DNN)", "Redes Neurais Convolucionais (CNN)", "Rede Neural Convolucional Baseada em Regiões (RCNN)", "IoT", "IP Camera", "offloading", "edge" e "fog". Esses termos foram selecionados por representarem ferramentas, arquiteturas e conceitos amplamente utilizados em pesquisas relacionadas à área. A inclusão de YOLO justifica-se pela sua ampla adoção na detecção de objetos, facilidade de uso e extensa documentação. Os termos DNN, CNN e RCNN referem-se a modelos de redes neurais comumente empregados em aplicações de visão computacional e reconhecimento de padrões.

Para estruturar a string de busca, foram utilizadas expressões booleanas com o objetivo de maximizar a recuperação de trabalhos pertinentes. Termos que representam conceitos distintos foram conectados pelo operador lógico AND, assegurando que todos estivessem presentes simultaneamente nos resultados. Já termos que expressam variações ou denominações correlatas — como CNN, DNN e RCNN — foram agrupados por meio do operador OR, permitindo a inclusão de publicações que utilizam abordagens diferentes na mesma linha de pesquisa deste trabalho.

### 2.1 Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Computing

Khan et al. (2019) [6] propuseram um sistema de chamada automatizada em salas de aula baseado em detecção de rosto e reconhecimento facial com CNN, utilizando o modelo Faster RCNN. Esse modelo possui dois fluxos: um para identificar regiões na imagem com maior chance de possuírem rostos e outro para reconhecer as faces, extraíndo características faciais. O sistema foi treinado com o dataset *Labeled Face in the World* (LFW) e conseguiu atingir 97,9% de acurácia na validação. Para gerar o banco de dados facial dos alunos, foram coletadas seis imagens por indivíduo e a chamada era feita com fotografias da turma enviadas ao sistema. A arquitetura incorporou computação de borda para reduzir a latência, com os serviços desenvolvidos em C#.Net e comunicação simples via HTTP. O sistema demonstrou boa eficiência, reconhecendo 30 de 35 faces

em sala e atingindo 94,6% de precisão média na detecção e 99,64% em avaliação com pares de rostos.

### 2.2 Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV

No trabalho de Khan et al. (2020) [7], é apresentado um sistema automatizado de chamada escolar baseado em reconhecimento facial, que utiliza câmeras instaladas nas salas para capturar imagens no início e no fim das aulas. As imagens são processadas pelo algoritmo YOLOv3, responsável pela detecção de rostos e, posteriormente, enviadas à Interface de Programação de Aplicações (API) de reconhecimento facial da *Microsoft Azure*, que identifica os alunos a partir de características faciais. O sistema inclui uma interface gráfica desenvolvida com *TKinter* e armazenamento local com *SQLite*, além de integração com a nuvem Azure para registro e identificação dos alunos. Após o reconhecimento, o sistema gera planilhas com os alunos presentes e ausentes e envia relatórios mensais por e-mail para pais, professores e os próprios alunos. Os testes realizados demonstraram alta precisão no reconhecimento, chegando a 100% em muitos casos, evidenciando a viabilidade de automatizar a chamada escolar e reduzir erros humanos.

### 2.3 Attendance System using Machine Learning-based Face Detection for Meeting Room Application

Muttaqin et al. (2020) [8] propõem um sistema de monitoramento automático de presença em tempo real para salas de reunião, baseado em reconhecimento facial com aprendizado de máquina. A arquitetura do sistema inclui três subsistemas: um *Front-End* em *Vue.js* para exibição dos dados, um *Back-End* com API REST para armazenamento e comunicação e um nó de captura e reconhecimento facial usando bibliotecas *OpenCV* e *Dlib*. As câmeras registram imagens no início e no fim das reuniões, processando-as com uma rede neural pré-treinada que extrai vetores faciais para comparação com o banco de dados. O sistema demonstrou desempenho elevado, com quase 100% de acurácia para rostos conhecidos, mas apresentou limitações em identificar corretamente rostos desconhecidos, com precisão variando entre 20% e 60%. Mesmo com tempo de resposta médio de dois segundos e distância ideal de dois metros, os autores destacam que o sistema ainda requer intervenção humana em alguns cenários para evitar falsos reconhecimentos.

### 2.4 Attendance System Based on Facial Recognition Using OpenCV

Rajesh et al. (2024) [10] propõem um sistema automatizado de controle de frequência escolar baseado em reconhecimento facial, desenvolvido com a biblioteca *OpenCV* em *Python*. O sistema foi testado com imagens de alunos capturadas sob diferentes condições de iluminação, ângulos e poses, demonstrando robustez na identificação facial, mesmo em cenários desafiadores. Os resultados indicaram alta precisão e eficiência com redução significativa no tempo e esforço administrativos associados ao controle de presença, sendo que os autores não mencionam valores numéricos. No entanto, o estudo carece de informações cruciais, como especificações

do *hardware* utilizado, número de participantes envolvidos e quantidade de imagens necessárias para o treinamento e funcionamento do sistema, o que limita a reproduzibilidade e a avaliação completa da proposta.

## 2.5 Discussão sobre as lacunas dos trabalhos relacionados

Apesar dos avanços significativos na utilização da visão computacional para sistemas de frequência escolar, persistem várias limitações em estudos anteriores. Por exemplo, [6] utiliza EC e algoritmos genéticos, mas depende do carregamento manual de fotografias por parte dos professores, o que introduz variabilidade na qualidade da imagem e continua dependendo do profissional para o utilizador. Do mesmo modo, em [7], requer acesso contínuo à Internet para processamento na nuvem e depende de grandes conjuntos de dados de imagens por aluno. O sistema carece de métricas de avaliação robustas e não consegue detectar ausências intermitentes. Em [8], o sistema optimiza a velocidade em ambientes restritos, mas mostra uma escalabilidade limitada devido ao número de participantes testados. A necessidade de configurações de câmera especializadas e a sensibilidade à distância. Já em [10], a falta de informações sobre a pesquisa é o que mais pesa sobre as lacunas do trabalho. Não se pode falar muita coisa sobre o trabalho se não oferece métricas para comparação com outras abordagens.

**Tabela 1: Comparação entre os trabalhos apresentados e as contribuições deste trabalho.**

Trabalho	Contribuição				
	1	2	3	4	5
Khan et al. (2019) [6]	X	X			
Khan et al. (2020) [7]	X			X	
Muttaqin et al. (2020) [8]	X				
Rajesh et al. (2024) [10]	X	X			X
Este trabalho	X	X	X	X	X

**1. Chamada em tempo real. 2 - Processamento dos dados em um *Raspberry Pi*. 3 - Reconhecimento facial em um dispositivo de borda. 4 - Avaliação do sistema com teste de estresse. 5 - Chamada automática a partir do reconhecimento facial.**

A Tabela 1 sintetiza os principais aspectos abordados pelos trabalhos analisados e os compara com as contribuições do presente estudo. Observa-se que todos os trabalhos contemplam a chamada em tempo real; contudo, a solução aqui proposta incorpora todas as contribuições identificadas nas pesquisas anteriores e, adicionalmente, introduz a realização de testes de estresse para avaliação do desempenho do sistema.

Além desses aspectos, o presente trabalho distingue-se por implementar uma solução de fácil gestão tanto do ponto de vista do software quanto da utilização prática, recorrendo a componentes de custo acessível e prontamente disponíveis no mercado. A escalabilidade do sistema foi validada empiricamente, evidenciando sua capacidade de adaptação a diferentes contextos de uso. Ademais, a

arquitetura proposta demonstra que é possível realizar reconhecimento facial de forma eficiente utilizando um dispositivo compacto e de baixo consumo energético, como o *Raspberry Pi*, o que reforça a viabilidade técnica e econômica da abordagem.

## 3 Arquitetura CLASSIC

Nesta seção, será apresentada a arquitetura do sistema, bem como cada um dos componentes que o forma. Além disso, serão exploradas as formas como o sistema realiza suas operações no intuito de realizar a frequência automatizada. A arquitetura do modelo é detalhada, destacando as diferentes camadas e como elas se interconectam para formar um sistema funcional.

### 3.1 Fluxo do Sistema

O sistema funciona em um processo de três fases. A primeira fase, denominada **aquisição de imagens**, é responsável pela captura de imagens, que servem de base para as fases seguintes. Em seguida, a segunda fase, **reconhecimento facial**, utiliza algoritmos sofisticados para determinar com precisão e autenticar as identidades dos alunos com base nas imagens capturadas. Na fase final, **registro de presenças**, os dados autenticados são sistematicamente registrados em uma base de dados, assegurando um controle de presenças preciso e eficiente. A Figura 1 mostra a relação abrangente entre essas fases, detalhadas nas seções seguintes.

**3.1.1 Aquisição de imagens.** A arquitetura do sistema inicia com a captura de imagens por uma câmera conectada diretamente à placa *Raspberry Pi*, utilizando um módulo de câmera comum. Antes de cada captura, o sistema verifica a presença de rostos, evitando o registro de imagens desnecessárias, desfocadas ou indistintas. Como a resolução tem uma quantidade significativa de *pixels*, torna-se adequada para recortar e identificar as faces dos alunos.

**3.1.2 Reconhecimento Facial.** A segunda fase pode ser executada através de duas modalidades operacionais distintas: 1) **reconhecimento local**, em que o dispositivo IoT processa a tarefa de forma independente, e 2) **remoto**, em que o dispositivo IoT delega a tarefa a um dispositivo EC computacionalmente mais poderoso.

O modo de reconhecimento local permite o reconhecimento facial diretamente no dispositivo IoT. Este módulo processa a imagem capturada para detectar e recortar o rosto ou rostos presentes na imagem e identificar os rostos, comparando-os com uma base de dados de entidades conhecidas. Este método permite que o sistema funcione independentemente de recursos externos, conforme a Figura 2, o que o torna particularmente vantajoso em conectividade de rede limitada ou baixo orçamento de implementação.

O modo de reconhecimento remoto é opcional e amplia as capacidades do sistema ao permitir a transferência de tarefas para um dispositivo EC com maior poder de processamento. No processo de inicialização, o sistema realiza uma varredura na rede local para identificar um servidor EC disponível, etapa necessária para a obtenção de parâmetros de rede – como endereços IP – e para o estabelecimento do canal de comunicação. Caso um EC seja encontrado, as requisições de reconhecimento passam a ser encaminhadas a ele, que executa todo o fluxo de processamento e devolve os resultados ao dispositivo IoT, responsável por notificar o usuário.

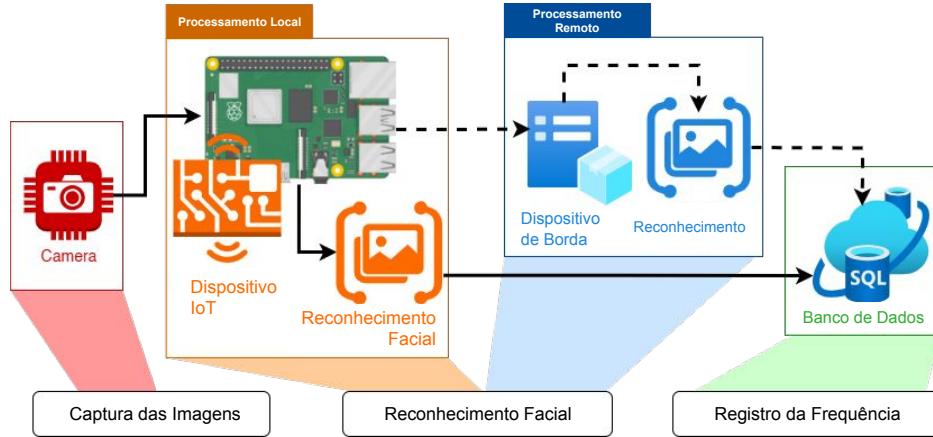


Figura 1: Arquitetura do sistema em nível Macro

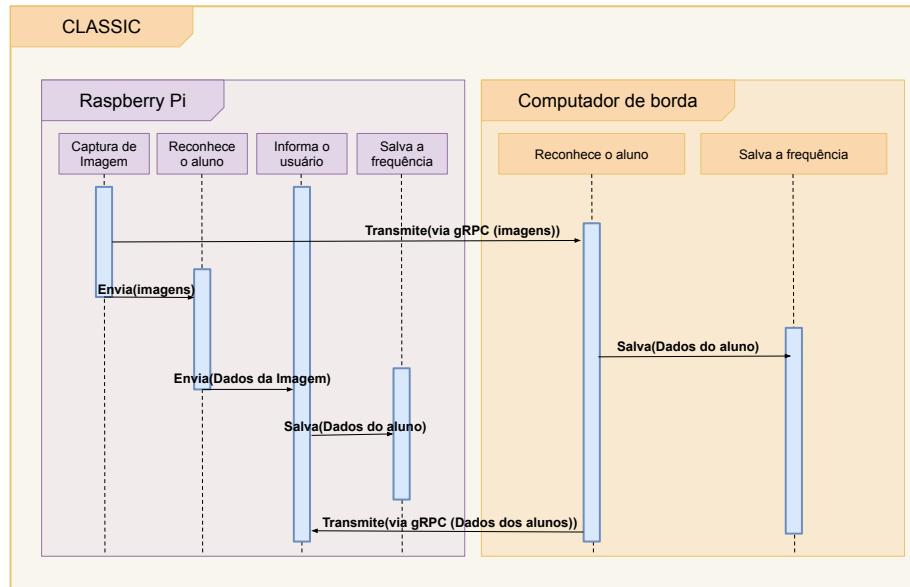


Figura 2: Diagrama de sequência com as ações do sistema.

Se, durante a inicialização, nenhum EC for detectado, o sistema configura-se automaticamente para operar exclusivamente no modo local, utilizando apenas o dispositivo IoT. Além disso, caso o EC seja desligado ou se torne indisponível em tempo de execução, o sistema alterna de forma transparente para o processamento local, garantindo a continuidade do funcionamento. Nesse sentido, as requisições de reconhecimento não são enviadas simultaneamente para ambos os dispositivos: o encaminhamento é definido dinamicamente de acordo com a disponibilidade do EC no momento da execução.

**3.1.3 Registro da Frequência.** Esta fase final consiste em registrar os resultados do processo de reconhecimento em uma base de dados. Tanto o dispositivo IoT como o EC estão equipados com um gestor de base de dados, que lhes permite registrar a presença localmente ou após processamento no EC. Cada registro inclui o identificador único do participante, a hora de reconhecimento e o estado de presença. A estrutura da base de dados assegura a integridade dos dados e permite uma consulta eficiente para gerar relatórios de assiduidade.



**Figura 3:** Configuração de hardware montada com *Raspberry Pi* e componentes, apresentando o resultado do reconhecimento, incluindo a identificação do aluno e o tempo de processamento.

### 3.2 Arquitetura de Hardware

O dispositivo IoT é um *Raspberry Pi 4B* com CPU Quad-Core Cortex-A72 e 4 GB de Memória de Acesso Aleatório (RAM), conectado por meio das portas Entrada/Saída de Propósito Geral (GPIO) a uma câmera OV5647 de 5 MP e a um ecrã LCD1602, responsável pela interação com o usuário e exibição do estado do sistema. Esses componentes foram fixados de forma segura em uma caixa personalizada impressa em 3D, conforme ilustrado na Figura 3. O sistema utiliza um computador com CPU Intel Core i5-12450H e 24 GB de RAM como dispositivo EC, selecionado pelo equilíbrio entre desempenho computacional e disponibilidade. Esse dispositivo processa imagens transferidas via *Wi-Fi*, executa o reconhecimento facial utilizando modelos avançados e retorna os resultados ao *Raspberry Pi*.

O terminal de usuário opera de forma autônoma segundo o seguinte fluxo: inicialmente, permanece em estado de espera até que uma pessoa se posicione diante do dispositivo; ao detectar a presença de um rosto, realiza a captura da imagem para o processo de reconhecimento; em seguida, apresenta no display a matrícula ou o nome do aluno identificado, juntamente com o tempo total de processamento da operação.

A interligação entre os componentes do sistema foi projetada para garantir comunicação confiável. Para isso, utilizou-se um cabo de 20 vias conectado diretamente ao *Raspberry Pi 4B*, que atua como unidade central de controle local. A comunicação com o módulo de câmera é realizada por meio de um cabo *flat* de 15 vias, conforme ilustrado na Figura 4.

### 3.3 Módulos de Software

A arquitetura emprega uma concessão modular que permite uma interação fluida entre componentes através da comunicação interna

ou da infraestrutura de rede existente, como ilustrado na Figura 5. O ambiente de desenvolvimento de software é baseado em Python 3.12, empregando bibliotecas e ferramentas amplamente utilizadas como: *OpenCV*, *Dlib*, *Pandas*, etc. Esta seção detalha as ferramentas, métodos e tecnologias utilizados para desenvolver e implementar as funcionalidades do sistema proposto.

**3.3.1 Módulo de Interface de Usuário.** Este módulo interage diretamente com os utilizadores para capturar imagens de alta qualidade através da câmera, preparando-as para processamento, seja instruindo quanto à posição para o registro fotográfico, seja para aguardar o resultado do reconhecimento. Como mencionado, mostra os resultados que são gerados pelo módulo de reconhecimento ao utilizador no visor LCD, fornecendo informações de retorno eficientes e em tempo real; por consequência, melhorando a experiência geral do utilizador.

**3.3.2 Módulo de Descoberta de Dispositivos.** Permite o reconhecimento automático e a comunicação sem descontinuidades entre dispositivos IoT e EC na mesma rede. É composto por dois componentes principais: o *Service Announcer*, alojado no dispositivo EC, e o *Service Finder*, alojado no dispositivo IoT. O *Announcer* transmite a disponibilidade e o endereço do dispositivo EC, enquanto o *Finder* configura a ligação em conformidade.

Durante o descarregamento, a comunicação entre os dispositivos é feita via Chamada de Procedimento Remoto do Google (gRPC)<sup>1</sup>, o que garante baixa latência e transferência eficiente de dados pela rede, além de permitir facilmente o descarregamento para processos de servidores desenvolvidos com linguagens mais eficientes no futuro [2]. O cliente estabelece comunicação com o dispositivo EC e descarrega tarefas para ele, enquanto o servidor trata os pedidos de reconhecimento recebidos, processa-os e devolve os resultados ao cliente, que informa ao usuário.

**3.3.3 Módulo de Reconhecimento.** Este módulo, que reside tanto nos dispositivos IoT como nos dispositivos de EC, opera com uma funcionalidade semelhante, identificando os alunos através das seguintes bibliotecas disponíveis: a) Biblioteca de Visão Computacional de Código Aberto (*OpenCV*)<sup>2</sup>, uma biblioteca de código aberto para visão computacional e processamento de imagem em tempo real, que detecta o rosto do aluno e prepara o formato do arquivo; em seguida, b) *DeepFace*<sup>3</sup>, uma biblioteca de um *framework* para análise e reconhecimento facial, que inclui suportes de vários modelos e tarefas, como a verificação facial e a detecção de emoções, é utilizada para reconhecer o aluno com base em imagens previamente existentes.

A biblioteca *DeepFace* permite ao usuário escolher dentre diversos parâmetros como se dará o processo de reconhecimento, desde qual modelo será utilizado para o recorte do rosto até qual será o modelo de extração de características das faces capturadas.

A biblioteca constrói um modelo de reconhecimento facial utilizando *TensorFlow* [1] e *Keras* [3], fazendo o download externo dos pesos pré-treinados dos modelos selecionados pelo usuário na invocação da função de reconhecimento. Realiza a detecção do rosto alinhando-os e depois recortando a área da face; depois, transforma

<sup>1</sup><https://grpc.io/>

<sup>2</sup><https://pypi.org/project/opencv-python/>

<sup>3</sup><https://pypi.org/project/deepface/>

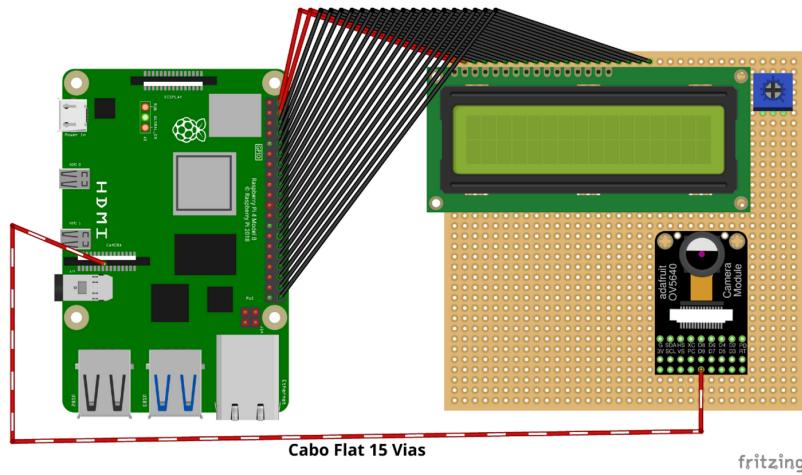


Figura 4: Projeção da conexão dos componentes dentro do terminal de usuário.

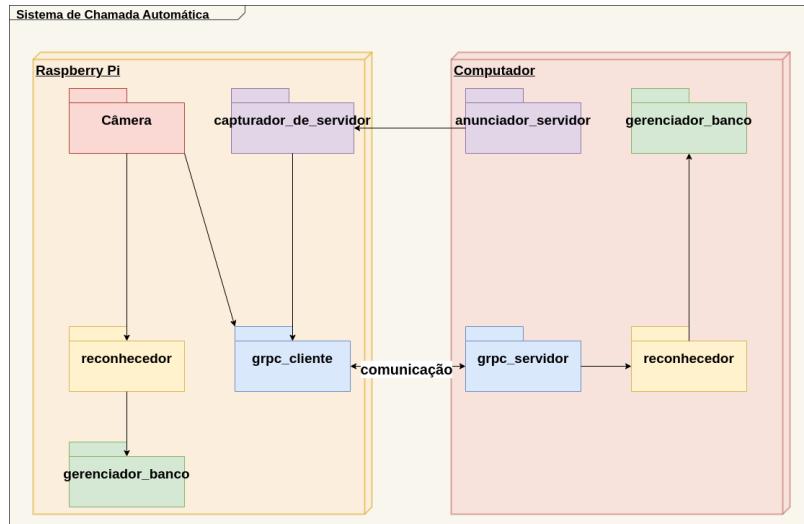


Figura 5: Diagrama de Módulos do sistema

pares de imagens em vetores com base no modelo carregado e, na sequência, calcula a distância entre essas representações utilizando uma métrica previamente definida também pelo usuário, por exemplo, Euclidiana. Por fim, determina se o par pertence à mesma pessoa com base em um limiar ideal que também pode ser definido pelo usuário. Todo esse processo é executado de forma automática em segundo plano [12].

Foram avaliados os modelos Detecção Multicaixa em Um Único Disparo (SSD) e ArcFace após testes iniciais em um conjunto de dados de 98 classes com 12 imagens cada, simulando várias condições para garantir a robustez no mundo real. Analisamos a precisão e o tempo de processamento considerando a combinação de modelos para detecção e reconhecimento de faces. Entre as configurações de modelos selecionadas, SSD + ArcFace apresentou uma precisão de 90% com um tempo médio de processamento de 1,89 segundos por imagem, quando executado no dispositivo EC. Além disso, os

modelos apresentaram um desempenho satisfatório em ambos os dispositivos em termos de tempo de resposta.

Selecionou-se este par de modelos para detecção e reconhecimento devido à sua compatibilidade imediata com o hardware e sistema utilizados. Na análise preliminar do sistema, calculou-se os custos de todas as combinações possíveis de modelos de detecção e reconhecimento, e os melhores resultados obtidos são apresentados na Tabela 2. Embora outros pares tenham apresentado maior precisão e tempos de execução semelhantes, sua implementação exigiu dependências de software que não puderam ser resolvidas dentro do prazo estabelecido para os experimentos e a tempo de serem apresentados nesta oportunidade.

**3.3.4 Gerenciamento de Banco de Dados.** Este módulo, que existe em ambos os dispositivos, registra os resultados do reconhecimento e permite um acompanhamento da assiduidade. O esquema da base

**Tabela 2: Pontuação atribuída à combinação Detector mais modelo**

Detector	Modelo	Acc. (%)	$\bar{x}$ (sec.)	S (sec.)
yolov8	ArcFace	98	1.98	1.1014
mtcnn	ArcFace	98	2.56	0.4736
retinaface	ArcFace	96	3.33	1.3341
dlib	ArcFace	94	2.21	0.5843
yolov8	VGG-Face	94	2.41	1.4694
mtcnn	VGG-Face	94	3.10	1.4378
yolov8	Facenet	93	1.89	0.1379
mtcnn	Facenet	93	2.51	0.4410
retinaface	Facenet	92	3.27	0.2986
retinaface	VGG-Face	92	3.84	0.4693
dlib	Dlib	91	2.07	0.5661
ssd	Facenet	90	1.86	1.9657
ssd	ArcFace	90	1.89	0.1214

de dados foi concebido para armazenar de forma eficaz as informações essenciais sobre alunos, turmas e eventos de assiduidade. Cada registro inclui o identificador único dos alunos, o *timestamp* de data/hora do reconhecimento e o estado (presente ou ausente). A base de dados relacional foi implementada em *SQLite*<sup>4</sup>, o que minimiza a sobrecarga de armazenamento, garantindo simultaneamente um desempenho suficiente dentro das necessidades da prova de conceito; além disso, é mais adequada para dispositivos IoT de baixa capacidade. Os resultados de qualquer um dos caminhos de reconhecimento são registrados pelo gestor da base de dados, garantindo um fluxo de trabalho eficiente e escalável. A Figura 6 apresenta o diagrama entidade-relacionamento da base de dados, mostrando as relações entre entidades.

## 4 Resultados e Experimentos

Os experimentos da arquitetura CLASSIC foram realizados em duas fases distintas, cada uma focando aspectos diferentes do desempenho do sistema:

**Implementação no Mundo Real.** O sistema foi testado como prova de conceito em um laboratório fechado, utilizando 14 voluntários. O dispositivo foi montado a uma altura de 1,5m, garantindo boa visibilidade do rosto dos participantes a uma distância de pelo menos um metro. O ambiente apresentava iluminação artificial e natural. Os participantes seguiram as instruções no ecrã para permanecerem imóveis durante a captura da imagem, bem como se estavam muito próximos do dispositivo. Essas informações são importantes, pois o dispositivo não possui tela de pré-visualização da imagem capturada. O sistema foi testado para reconhecimento local no *Raspberry Pi* e reconhecimento assistido por EC utilizando um dispositivo computacional mais potente. Independentemente do local de processamento, todas as imagens foram armazenadas para análise posterior na Fase 2.

**Teste de Estresse e Avaliação Performática.** As imagens recolhidas durante a Fase 1 foram utilizadas em testes de estresse

computacional para avaliar o desempenho do sistema em diferentes condições. O mesmo par detector-modelo foi utilizado no *Raspberry Pi* e no dispositivo de borda da rede para comparar os resultados. Os testes de estresse foram realizados simulando vários clientes a fazer pedidos em simultâneo para avaliar a escalabilidade.

### 4.1 Implementação no Mundo Real

Para avaliar o desempenho do sistema no cenário do mundo real, o sistema foi testado nas duas configurações apresentadas na Seção 3.1.2: 1) reconhecimento local usando o dispositivo IoT, que executou todo o *pipeline* de reconhecimento, e 2) reconhecimento assistido pela borda, onde a tarefa de reconhecimento foi descarregada para o dispositivo EC. Os participantes foram orientados a interagir com o ecrã e a seguir as instruções. Por exemplo, se não fosse detectado nenhum rosto, era apresentada uma mensagem a pedir aos participantes que ajustassem a sua posição.

O CLASSIC apresentou um tempo médio para processamento local no dispositivo IoT de 4,5 segundos por reconhecimento. Por outro lado, o processamento assistido pelo EC obteve um tempo de 1,97 segundos, 56,2% menor que o tempo do IoT. Tendo em conta a precisão do reconhecimento, em ambas as configurações, o sistema reconheceu corretamente 12 dos 14 participantes, obtendo uma precisão de 85,71%. De acordo com o retorno avaliativo dos participantes, o sistema apresentou uma boa capacidade de resposta e uma resposta clara através do ecrã. No entanto, a utilização de acessórios, como óculos e chapéus, provocou ocasionalmente erros de reconhecimento devido ao encandeamento ou ao obscurecimento de características faciais, que levaram a uma identificação incorreta.

### 4.2 Teste de Estresse e Avaliação Performática

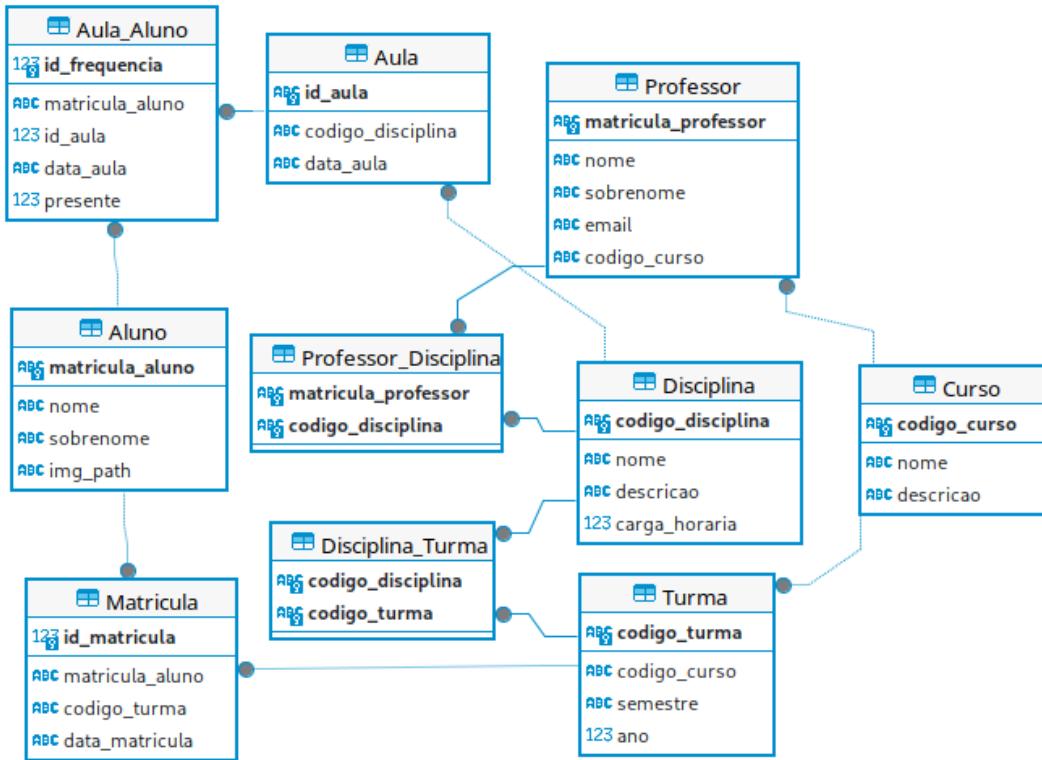
A segunda fase simulou cenários de elevada utilização para avaliar a escalabilidade. Observando a infraestrutura da universidade local, a experiência simulou até 25 salas de aula com exatamente 50 alunos cada e que funcionam em simultâneo. Um segundo computador simulou pedidos ao servidor, como se fossem dispositivos em sala de aula, utilizando imagens reais recolhidas durante a Fase 1. Nesta experiência, foi utilizado apenas o reconhecimento que foi realizado na borda da rede.

Cada uma das salas de aula simuladas, aqui denominadas *threads*, processou 50 imagens por execução, repetidas 30 vezes. A Tabela 3 resume as métricas de desempenho, mostrando o comportamento estatístico da precisão do sistema e dos tempos de reconhecimento à medida que a carga aumentava: média ( $\bar{x}$ ), mediana ( $\tilde{x}$ ), desvio padrão ( $\sigma$ ) e quartis ( $q_1, q_3$ ). Apesar do aumento da carga, o sistema manteve uma precisão próxima de 85% em todos os cenários de aulas simultâneas, o que está de acordo com os resultados obtidos na Seção 4.1. Os resultados sugerem que o CLASSIC apresentou com sucesso um tempo total de reconhecimento inferior a 5 segundos ao processar cinco turmas em simultâneo.

### 4.3 Limitações do Sistema CLASSIC

Embora o CLASSIC tenha apresentado resultados promissores nos cenários avaliados, algumas limitações foram identificadas durante os experimentos e merecem destaque para orientar trabalhos futuros.

<sup>4</sup><https://www.sqlite.org/>



**Figura 6:** Diagrama entidade-relacionamento do banco de dados projetado para os experimentos

**Tabela 3: Resumo dos resultados nos testes de estresse computacional.**

Threads	$\bar{x}$ (s)	$q_1$ (s)	$\tilde{x}$ (s)	$q_3$ (s)	$\sigma$ (s)	Acc. (%)
1	1.39	1.23	1.31	1.43	0.41	84.06
5	3.69	3.09	3.31	3.78	1.15	84.00
10	7.95	6.47	7.35	8.75	2.29	84.00
15	10.31	9.09	9.51	11.65	2.30	84.00
20	28.32	21.46	25.26	27.96	12.10	84.00
25	34.41	29.15	32.27	37.91	16.42	84.23

Em primeiro lugar, a precisão média obtida ( $\approx 85\%$ ) ficou aquém de valores superiores a 90% que já haviam sido reportados em testes controlados com o mesmo par detetor–modelo. Essa diferença pode ser atribuída, em parte, à variabilidade introduzida pelo uso de acessórios como óculos e chapéus, que provocaram encandeamento ou ocultação de características faciais relevantes, prejudicando o reconhecimento.

Outra limitação refere-se ao número reduzido de participantes na fase de implementação no mundo real, restrito a 14 voluntários do laboratório. Apesar de representarem perfis distintos, esse conjunto é significativamente menor que o encontrado em turmas reais com 40–50 estudantes, o que restringe a generalização dos resultados e a avaliação de aspectos como o reconhecimento simultâneo de múltiplos rostos no enquadramento.

Além disso, o ambiente de teste apresentou iluminação mista (natural e artificial), porém predominantemente controlada e interna. Embora a condição seja representativa de muitos espaços fechados, a ausência de cenários com iluminação mais desafiadora, como forte luz solar direta ou ambientes mal iluminados, limita a compreensão do comportamento do sistema em condições adversas.

Outra questão observada diz respeito à necessidade de os alunos fornecerem previamente imagens pessoais para o cadastro no sistema. Embora essa etapa seja fundamental para o funcionamento do reconhecimento, ela pode representar um ponto de atenção em termos de logística e aceitação, especialmente em ambientes com grande número de usuários.

Adicionalmente, na configuração atual, o sistema realiza o reconhecimento de um aluno por vez. Embora essa abordagem simplifique o processamento e reduza a possibilidade de erros, trabalhos futuros poderão explorar estratégias para ampliar essa capacidade e permitir o reconhecimento simultâneo de múltiplos indivíduos.

Por fim, o desempenho observado depende diretamente da escolha do detetor e do modelo de reconhecimento facial. Embora a arquitetura proposta seja flexível e permita a substituição desses componentes, os testes atuais foram restritos a um conjunto específico, que pode não ser o mais robusto para todos os contextos de uso.

## 5 Conclusão

Este artigo apresentou o CLASSIC, um sistema inteligente de controle de frequência escolar que integra tecnologias de IoT, EC e reconhecimento facial, com o objetivo de automatizar o rastreamento de presença em ambientes educacionais. Contudo, a forma como o sistema está disposto e projetado permite o uso em ambientes empresariais para, por exemplo, controle de funcionários. A arquitetura do sistema combina um dispositivo *Raspberry Pi* de baixo consumo energético e alta portabilidade com uma estrutura modular que suporta modos de operação local e remoto. A utilização de visão computacional no próprio dispositivo permite otimizar o tempo dos profissionais de educação e gerar dados úteis à instituição, além de possibilitar aplicações práticas para diversos desafios do cotidiano escolar.

O sistema realiza o reconhecimento facial com precisão de até 85%, mesmo em cenários com condições variáveis de iluminação, ângulos e múltiplas requisições simultâneas. Em testes com até quinze alunos, registrando presença ao mesmo tempo, o desempenho se manteve satisfatório, com tempos de resposta adequados à realidade da sala de aula. A estratégia de *offloading* computacional para dispositivos de borda aumentou significativamente a escalabilidade do sistema, reduzindo a latência e evitando sobrecarga do dispositivo principal. Essa abordagem se mostrou especialmente eficaz quando o número de *workers* gRPC ultrapassou cinco, conforme observado nos testes.

Os resultados experimentais confirmam a viabilidade da solução para implantação no mundo real, com robustez diante de cenários de alto estresse computacional e boa precisão nas tarefas de reconhecimento facial. A arquitetura modular ainda oferece flexibilidade para futuras extensões, como a adoção de novos algoritmos de reconhecimento ou protocolos de comunicação mais avançados, permitindo ao sistema se adaptar à evolução das demandas educacionais.

Como trabalhos futuros, sugere-se a realização de experimentos com uma rede dedicada ao serviço e com maior capacidade de tráfego, o que não ocorreu nos testes atuais. Também se recomenda ampliar a base de participantes para avaliar o comportamento do sistema em grupos maiores e mais heterogêneos, bem como incluir cenários com condições de iluminação mais desafiadoras. Além disso, é relevante estudar estratégias para permitir o reconhecimento simultâneo de múltiplos alunos, visto que a versão atual processa um indivíduo por vez, e explorar alternativas para otimizar o cadastro, de forma a reduzir a necessidade de coleta manual de imagens pessoais. Outro ponto a ser explorado é a melhoria da experiência do usuário, incluindo maior acessibilidade, como a implementação de áudio descrição para deficientes visuais e a adição de uma segunda câmera para melhor enquadramento de cadeirantes. Por fim, considerando que a biblioteca DeepFace utilizada no projeto é flexível, investigações futuras poderão avaliar abordagens de distribuição de processamento, por exemplo, delegando ao dispositivo de borda a extração dos vetores de características faciais e deixando para o dispositivo IoT apenas a etapa final de cálculo de similaridade.

## AGRADECIMENTOS

A todo o time do EngineLab da Universidade Federal do Ceará no Campus de Crateús, sobretudo meus caros amigos: Gabriel Rudan, Letícia Torres, Raylander Marques e Zairo Bastos.

## REFERÊNCIAS

- [1] Martín Abadi et al. “{TensorFlow}: a system for {Large-Scale} machine learning”. Em: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016, pp. 265–283.
- [2] Filipe B. de Matos, Paulo Rego, e Fernando Trinta. “An Empirical Study about the Adoption of Multi-language Technique in Computation Offloading in a Mobile Cloud Computing Scenario”. Em: *Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER*, INSTICC. SciTePress, 2021, pp. 207–214. ISBN: 978-989-758-510-4. doi: 10.5220/0010437802070214.
- [3] Francois Chollet. *Deep learning with Python*. simon e schuster, 2021.
- [4] Debashis De. *Mobile cloud computing: architectures, algorithms and applications*. CRC Press, 2016.
- [5] Gabriel Yuri Hosaki e Douglas Francisco Ribeiro. “Deep learning: ensinando a aprender”. Em: *RGE - Revista de Gestão e Estratégia* 1.3 (2021), pp. 36–50.
- [6] Muhammad Zeeshan Khan et al. “Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Computing”. Em: *IEEE Access* 7 (2019), 72622 – 72633. doi: 10.1109/ACCESS.2019.2918275.
- [7] Sikandar Khan, Adeel Akram e Nighat Usman. “Real Time Automatic Attendance System for Face Recognition Using Face API and OpenCV”. Em: *Wireless Personal Communications* 113.1 (2020), 469 – 480. doi: 10.1007/s11277-020-07224-2.
- [8] Rahmat Muttaqin, Syifaul Fuada, Eueung Mulyana et al. “Attendance System using Machine Learning-based Face Detection for Meeting Room Application”. Em: *International Journal of Advanced Computer Science and Applications* 11.8 (2020).
- [9] Enivaldo Sousa Paiva e Otálio Rodrigues da Silva. “O USO DO TEMPO PEDAGÓGICO NUMA ESCOLA DE TEMPO INTEGRAL DO ENSINO MÉDIO”. Em: *VI Congresso Nacional de Educação* (2020).
- [10] Thota Radha Rajesh et al. “Attendance System Based on Facial Recognition Using OpenCV”. Em: *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. 2024, pp. 1529–1535. doi: 10.1109/ICSCSS60660.2024.10624932.
- [11] Bruno P Santos et al. “Internet das coisas: da teoria à prática”. Em: *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* 31 (2016), p. 16.
- [12] Sefik Ilkin Serengil e Alper Ozpinar. “Lightface: A hybrid deep face recognition framework”. Em: *2020 innovations in intelligent systems and applications conference (ASYU)*. IEEE. 2020, pp. 1–5.