

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA, E TECNOLOGIA DA PARAÍBA
CAMPUS CAMPINA GRANDE
COORDENAÇÃO DE CURSO SUPERIOR DE ENGENHARIA FE COMPUTAÇÃO

ISAAC ANTONIO SILVA BARBOSA
JOÃO PEDRO DE SOUSA CORREA

**DESENVOLVIMENTO DE SOFTWARE DE RECONHECIMENTO
FACIAL PARA CONTABILIZAR PRESENÇA DE ALUNO EM SALA DE
AULA**

Campina Grande - PB
2024

ISAAC ANTONIO SILVA BARBOSA
JOÃO PEDRO DE SOUSA CORREA

**DESENVOLVIMENTO DE SOFTWARE DE RECONHECIMENTO
FACIAL PARA CONTABILIZAR PRESENÇA DE ALUNO EM SALA DE
AULA**

Análise de software desenvolvido como parte do Trabalho de Conclusão de Curso em Engenharia de Computação pelo Instituto Federal da Paraíba, IFPB. Este trabalho tem como objetivo a criação e análise de um software desenvolvido pelos autores. A análise abrange os aspectos técnicos e funcionais do software, bem como a sua implementação e resultados obtidos

Orientador: Prof. Dr Katyusco de Farias Santos.

Campina Grande - PB
2024

B238d Barbosa, Isaac Antonio Silva

Desenvolvimento de software de reconhecimento facial para contabilizar presença de aluno em sala de aula. / Isaac Antonio Silva Barbosa, João Pedro de Sousa Correa. - Campina Grande, 2024.

64 f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Bacharelado em Engenharia de Computação) - Instituto Federal da Paraíba, 2024.

Orientador: Prof. Dr. Katyusco de Farias Santos.

1. Reconhecimento biométrico - facial 2. Automação em sala de aula - Controle de presença 3. Inteligência artificial - IA. I. Correa, João Pedro de Sousa II. Santos, Katyusco de Farias III. Título.

ISAAC ANTONIO SILVA BARBOSA
JOÃO PEDRO DE SOUSA CORREA

**DESENVOLVIMENTO DE SOFTWARE DE RECONHECIMENTO
FACIAL PARA CONTABILIZAR PRESENÇA DE ALUNO EM SALA DE
AULA**

Trabalho de Conclusão de Curso apresentado
ao curso de **Engenharia de Computação** do
Instituto Federal da Paraíba, IFPB, como
requisito parcial à obtenção do título de
Bacharel em. **Engenharia de Computação**.

Local, ____ de _____ de ____.

Banca Examinadora:

Prof. Katyusco de Farias Santos
Instituto Federal da Paraíba, IFPB

Prof. Danyllo Wagner Albuquerque
Instituto Federal da Paraíba, IFPB

Prof. Paulo Ribeiro Lins Junior
Instituto Federal da Paraíba, IFPB

A todos que se tornaram pilares durante esta jornada, dedicamos este trabalho com profunda gratidão. Aos nossos pais, cuja fé em nosso potencial foi constante, tornando possível cada passo. Aos amigos, cujas risadas e apoio fizeram os desafios mais suportáveis. E às nossas parceiras, cuja companhia nos ancorou em meio ao caos. Este trabalho é uma teia de pequenas contribuições, tecida por aqueles que acreditaram e caminharam ao nosso lado.

AGRADECIMENTOS

Gostaríamos de expressar nossa sincera gratidão a todos que contribuíram para a realização deste trabalho.

Aos nossos pais, que mesmo à distância ou ao nosso lado, ofereceram um apoio incondicional e constante, proporcionando a força necessária para seguir em frente. Aos amigos, cujo apoio e incentivo trouxeram leveza aos momentos mais difíceis.

Agradecemos também às nossas parceiras, cujo carinho e paciência foram essenciais para mantermos o equilíbrio ao longo desta jornada.

Por fim, um agradecimento especial ao nosso professor e orientador, Katusco de Farias Santos, cuja inspiração foi fundamental para a escolha do tema deste trabalho.

Este projeto é fruto de um esforço coletivo, e somos profundamente gratos a todos que nos apoiaram ao longo deste caminho.

“Ninguém é igual a ninguém. Todo o ser humano é um estranho ímpar.”

Carlos Drummond de Andrade

RESUMO

Este trabalho apresenta o desenvolvimento de um software de reconhecimento facial destinado a automatizar o controle de presença de alunos em sala de aula, utilizando dispositivos móveis para captura de imagens. O contexto se insere na necessidade de otimizar o tempo dos professores durante o processo de chamada e minimizar o desgaste físico e emocional associado ao método tradicional de contabilizar presenças, principalmente em turmas grandes. O objetivo principal é criar um sistema eficiente, baseado em imagens, que substitua a chamada manual, garantindo maior precisão e agilidade no controle de presença. O método utilizado envolveu duas fases de desenvolvimento: uma primeira versão utilizando vídeos para reconhecimento facial e uma segunda versão baseada em imagens estáticas, após a primeira abordagem demonstrar limitações em termos de precisão e tempo de processamento. Os resultados indicaram que o uso de imagens estáticas apresentou uma significativa melhoria no desempenho do sistema, tanto em termos de precisão quanto de eficiência no tempo de processamento. Conclui-se que a solução proposta é viável e eficaz para o controle de presença automatizado em sala de aula, com potencial para futuras expansões e integração com outros sistemas educacionais.

Palavras-chave: Reconhecimento facial; Desenvolvimento web; Sala de aula; Presença de alunos; Encoding.

ABSTRACT

This work presents the development of facial recognition software designed to automate control of student presence in the classroom, using mobile devices to capture images. The context is part of the need to optimize teachers' time during the call process and minimize the physical and emotional strain associated with the traditional method of counting attendance, especially in large classes. The main objective is to create an efficient system, based on images, that replaces manual calling, ensuring greater precision and agility in attendance control. The method used involved two development phases: a first version using videos for facial recognition and a second version based on static images, after the first approach demonstrated limitations in terms of accuracy and processing time. The results indicated that the use of static images presented a significant improvement in system performance, both in terms of accuracy and efficiency in processing time. It is concluded that the proposed solution is viable and effective for automated attendance control in the classroom, with potential for future expansion and integration with other educational systems.

Keywords: Facial recognition; Web development; Classroom; Presence of students; Encoding.

LISTA DE FIGURAS

Figura 1 – Diagrama de casos de uso do sistema.....	25
Figura 2 – Fluxo da função recognition.....	30
Figura 3 – Tela inicial da aplicação: Opções de login e cadastro.....	31
Figura 4 – Tela de login.....	32
Figura 5 – Tela de cadastro para novos usuários.....	33
Figura 6 – Tela de termos de consentimento.....	33
Figura 7 – Tela de criação de alunos.....	34
Figura 8 – Tela de criação de alunos preenchida.....	35
Figura 9 – Tela de criação de turma.....	35
Figura 10 – Tela de criação de turma preenchida.....	36
Figura 11 – Tela de visualização de turmas.....	36
Figura 12 – Tela de visualização de alunos.....	37
Figura 13 – Tela de reconhecimento.....	37
Figura 14 – Tela de reconhecimento preenchida.....	38
Figura 15 – Tela de verificação da lista de alunos.....	38
Figura 16 – Tela de chamadas realizadas.....	39
Figura 17 – Pré-visualização do CSV da chamada.....	39
Figura 18 – Pasta gerada para os testes.....	41
Figura 19 – Foto de registro do aluno 1.....	42
Figura 20 – Foto de registro do aluno 2.....	42
Figura 21 – Captura de tela do vídeo teste.....	43
Figura 22 – Alunos cadastrados.....	43
Figura 23 – Turma cadastrada.....	44
Figura 24 – Checklist com resultado do reconhecimento.....	45
Figura 25 – CSV de presença de alunos.....	46
Figura 26 – Novo Fluxo da Função recognition.....	48
Figura 27 – Tela de reconhecimento com imagens.....	50
Figura 28 – Tela de reconhecimento com imagens preenchida.....	50
Figura 29 – Diagrama de casos de uso da versão piloto 2.0 do sistema.....	51
Figura 30 – Resultados obtidos do vídeo 1.....	53
Figura 31 – Resultados obtidos do vídeo 2.....	54
Figura 32 – Resultados obtidos do vídeo 3.....	55
Figura 33 – Resultados obtidos do vídeo 4.....	56
Figura 34 – Tempo de Processamento por Vídeo na Versão Piloto 1.0 do Sistema.....	57
Figura 35 – Resultados obtidos das imagens capturadas do Vídeo 1.....	57
Figura 36 – Resultados obtidos das imagens capturadas do Vídeo 2.....	58
Figura 37 – Resultados obtidos das imagens capturadas do Vídeo 3.....	59
Figura 38 – Resultados obtidos das imagens capturadas do Vídeo 4.....	60

Figura 39 – Tempo de Processamento por Conjunto de imagens na Versão Piloto 2.0 do Sistema.....	61
Figura 40 – Comparação dos Tempos de Processamento entre as Versões do Sistema.....	62
Figura 41 – Comparação da Eficácia entre as versões do sistema.....	63

LISTA DE ABREVIATURAS E SIGLAS

TCC	Trabalho de Conclusão do Curso
LGPD	Lei Geral de Proteção de Dados
APIs	Interface de programação de aplicações
CRUD	Criar, Ler, Atualizar e Excluir
CSV	Valores Separados por Vírgula
HOG	Histograma de gradientes orientados
CNN	Rede neural convolucional

SUMÁRIO

1.0. INTRODUÇÃO.....	14
1.1 OBJETIVO GERAL.....	15
1.2. OBJETIVOS ESPECÍFICOS.....	15
1.3. PROJETO RELACIONADO.....	15
1.3.1. Inteligência artificial: a importância do reconhecimento facial na educação:.....	16
2.0. FUNDAMENTAÇÃO TEÓRICA.....	17
2.1. O QUE É A CHAMADA DE ALUNOS.....	17
2.2. RECONHECIMENTO FACIAL.....	17
2.3. DLIB.....	18
2.4. EMBEDDINGS.....	18
2.5. FACE RECOGNITION 1.3.0.....	18
2.6. CSV.....	19
2.7. PROCESSO DE ORGANIZAÇÃO.....	19
2.7.1. Git e GitHub.....	19
2.7.2. Figma.....	20
2.8. DOCKER.....	20
2.9. PRIVACIDADE E ÉTICA SOB AS LGPD.....	20
3.0. METODOLOGIA.....	22
3.1. RESTRIÇÕES DO SISTEMA:.....	22
3.1.1. Requisitos funcionais.....	23
3.1.2. Requisitos não funcionais:.....	24
3.2. PESQUISA BIBLIOGRÁFICA.....	25
3.2.1 Front-end.....	26
3.2.2. Back-end.....	27
3.3. DESENVOLVIMENTO DA VERSÃO PILOTO 1.0 DO SISTEMA.....	28
3.3.1. Desenvolvimento do front-end.....	29
3.3.2. Desenvolvimento do back-end.....	29
3.4. DEMONSTRAÇÃO DA VERSÃO PILOTO 1.0 DO SISTEMA.....	31
3.4.1. Login e Cadastro.....	31
3.4.2. Termos.....	33
3.4.3. Criação de Alunos.....	34
3.4.4. Criação de Turmas.....	35
3.4.5. Turmas e Alunos.....	36
3.4.6. Reconhecimento.....	37
3.4.7. Confirmação da Lista de Alunos Presentes.....	38
3.4.8. Chamadas Realizadas.....	39
3.5. USO DE CONTÊINER.....	40

3.6. CONFIGURAÇÃO DO HARDWARE UTILIZADO PARA OS TESTES.....	40
3.7. MÉTRICAS UTILIZADAS NOS TESTES.....	40
3.8. TESTE DA VERSÃO PILOTO 1.0 DO SISTEMA.....	42
3.9. TESTE DA VERSÃO PILOTO 1.0 DO SISTEMA EM AMBIENTE REAL.....	46
3.9.1 Consentimento e Participação.....	47
3.9.2 Resultado do teste da versão piloto 1.0 do sistema.....	47
3.10. DESENVOLVIMENTO DA VERSÃO PILOTO 2.0 DO SISTEMA.....	48
3.11. TESTE DA VERSÃO PILOTO 2.0 DO SISTEMA EM AMBIENTE REAL.....	51
3.12. LIMITAÇÕES DOS TESTES.....	52
4.0. RESULTADOS E DISCUSSÕES.....	53
4.1 RESULTADOS DOS TESTES DA VERSÃO PILOTO 1.0 DO SISTEMA.....	53
4.2 RESULTADOS DOS TESTES DA VERSÃO PILOTO 2.0 DO SISTEMA.....	57
4.3 ANÁLISE DOS RESULTADOS.....	61
5.0. CONCLUSÃO E TRABALHOS FUTUROS.....	64

1.0. INTRODUÇÃO

O avanço das tecnologias de reconhecimento facial tem promovido uma revolução em diversas áreas, incluindo a educação. No contexto acadêmico, a contagem manual da presença de alunos em sala de aula ainda é um processo comum, mas é marcado por ineficiências. Esse cenário oferece uma oportunidade única para o desenvolvimento de soluções tecnológicas que otimizem o tempo e melhorem a precisão desse processo, especialmente em turmas grandes, onde o desgaste físico e emocional dos professores é mais evidente.

O presente trabalho, intitulado "Desenvolvimento de Software de Reconhecimento Facial para Contabilizar Presença de Aluno em Sala de Aula", tem como tema a aplicação de tecnologias de reconhecimento facial para automatizar o controle de presença dos alunos. A pesquisa está inserida no contexto da inovação educacional, com foco na melhoria da gestão do tempo em sala de aula e na redução dos impactos físicos e emocionais sobre os professores.

A pesquisa delimita-se ao desenvolvimento e à análise de um sistema de reconhecimento facial voltado exclusivamente para contabilizar a presença de alunos em sala de aula. O software será projetado para uso em dispositivos móveis, facilitando a captura de imagens no ambiente escolar. Além disso, o estudo aborda a implementação de medidas de segurança para proteger os dados biométricos dos alunos, conforme as exigências da Lei Geral de Proteção de Dados (LGPD).

A escolha por desenvolver um sistema de reconhecimento facial é justificada pela necessidade de otimizar o tempo dos professores em sala de aula, que tradicionalmente é gasto em tarefas repetitivas, como a chamada de alunos. Além disso, o uso dessa tecnologia proporciona maior precisão na contabilização de presenças, reduzindo a possibilidade de erros. O professor Katysco de Farias Santos, ao longo de suas atividades, apontou o desgaste físico decorrente da repetição dessa tarefa, o que reforça a necessidade de uma solução prática e eficiente para mitigar esses problemas.

A metodologia adotada neste trabalho envolve o desenvolvimento do sistema em duas etapas. Primeiramente, foi realizada uma pesquisa bibliográfica para a escolha das tecnologias adequadas para a implementação da aplicação. Em seguida, o desenvolvimento do sistema foi dividido em duas versões. A versão piloto 1.0 utilizou vídeos para o reconhecimento facial,

mas após testes, optou-se pela versão piloto 2.0, que utiliza imagens estáticas para garantir maior precisão. O software será testado em ambiente real com professores e alunos, sendo avaliados aspectos como o tempo de processamento e a precisão das detecções.

Este trabalho está dividido em cinco capítulos. O primeiro capítulo apresenta a introdução ao tema e os objetivos da pesquisa. O segundo capítulo aborda a fundamentação teórica, explorando os principais conceitos relacionados ao reconhecimento facial e à aplicação dessa tecnologia na educação. O terceiro capítulo detalha a metodologia utilizada para o desenvolvimento e implementação do sistema. O quarto capítulo apresenta os resultados dos testes realizados com as duas versões do software, bem como a análise desses resultados. Por fim, o quinto capítulo traz as conclusões do trabalho e sugestões para pesquisas futuras.

1.1 OBJETIVO GERAL

O objetivo deste Trabalho de Conclusão de Curso é desenvolver uma aplicação de reconhecimento facial para automatizar a chamada de presença em sala de aula, utilizando um dispositivo móvel para capturar imagens.

1.2. OBJETIVOS ESPECÍFICOS

Para atingir esse objetivo principal, estabelecemos os seguintes objetivos específicos:

- Desenvolver um sistema web responsivo que se adapte a diferentes dispositivos móveis, proporcionando uma interface de usuário otimizada para a coleta e gerenciamento de dados
- Implementar funcionalidades que permitam o upload e a análise de imagens capturadas por dispositivos móveis para o reconhecimento facial.
- Garantir a persistência e integridade dos dados de usuários, alunos, turmas e registros de presença em uma base de dados segura.

1.3. PROJETO RELACIONADO

Nesta seção, é apresentado um projeto que visa o desenvolvimento de software para a realização de chamadas utilizando reconhecimento facial.

1.3.1. Inteligência artificial: a importância do reconhecimento facial na educação:

O artigo de Monteiro (2020) explora o impacto do reconhecimento facial na modernização do controle de presença nas instituições de ensino do Brasil. Utilizando a linguagem de programação Python e a biblioteca OpenCV, os autores propõem um sistema automatizado que substitui métodos tradicionais de chamada, visando aumentar a eficiência e a segurança no ambiente escolar. O estudo demonstra a viabilidade e os benefícios dessa tecnologia no contexto educacional brasileiro. No entanto, embora o estudo tenha idealizado o desenvolvimento do software, ele não foi implementado na prática. Além disso, a solução proposta depende do uso de câmeras e computadores com alto poder de processamento, ignorando a possibilidade de utilização de dispositivos móveis. Isso limita sua acessibilidade e praticidade, especialmente em ambientes escolares, onde a mobilidade e a simplicidade de operação são essenciais.

2.0. FUNDAMENTAÇÃO TEÓRICA

Esta fundamentação teórica explora os principais conceitos que sustentam o desenvolvimento do sistema proposto, abordando temas como o reconhecimento facial, a integração de tecnologias para a gestão de turmas, o processamento de imagens e os algoritmos de reconhecimento facial. Além disso, são discutidas as considerações éticas e de privacidade associadas ao uso dessas tecnologias. Esse embasamento teórico é essencial para compreender os conceitos e as escolhas tecnológicas que orientam este projeto, proporcionando uma base sólida para o entendimento do sistema desenvolvido.

2.1. O QUE É A CHAMADA DE ALUNOS

Para compreender a motivação e a importância do desenvolvimento deste projeto, é essencial entender o conceito de chamada de alunos. A chamada é uma atividade administrativa realizada pelos professores em sala de aula, cuja finalidade é contabilizar a frequência dos alunos. Esse controle é fundamental tanto para o acompanhamento do desempenho acadêmico quanto para o cumprimento de requisitos institucionais, como a exigência de presença mínima para aprovação em disciplinas.

Tradicionalmente, essa tarefa é realizada de forma manual, seja por meio de chamada oral, onde o professor menciona o nome de cada aluno, ou por listas de presença, onde os próprios alunos escrevem seus nomes e repassam a lista. Esses métodos, embora comuns, podem ser demorados e propensos a erros, especialmente em turmas numerosas. Erros de preenchimento, esquecimentos e falhas na comunicação podem comprometer a precisão dos registros.

2.2. RECONHECIMENTO FACIAL

O reconhecimento facial é uma tecnologia revolucionária que utiliza algoritmos de aprendizagem de máquina para identificar indivíduos com base em características únicas de seus rostos. Estudos pioneiros, como os conduzidos por Turk e Pentland (“Eigenfaces for Recognition” 1991), enfatizam a importância da análise meticulosa de traços faciais distintivos, como configuração dos olhos, nariz e boca, na eficácia desses sistemas. Essa tecnologia é amplamente aplicada em segurança, marketing e agora também na educação, demonstrando sua relevância e eficácia em diversos contextos. Mais recentemente, sua aplicação expandiu-se para áreas como gestão de eventos, controle de acesso em espaços

públicos e até mesmo no setor bancário, reforçando a importância de desenvolver sistemas precisos e confiáveis que possam se adaptar a diferentes condições operacionais.

2.3. DLIB

Dlib é um kit de ferramentas C++ moderno contendo algoritmos de aprendizado de máquina e ferramentas para criar software complexo em C++ para resolver problemas do mundo real. Ele é usado tanto na indústria quanto na academia em uma ampla gama de domínios, incluindo robótica, dispositivos embarcados, telefones celulares e grandes ambientes de computação de alto desempenho. O licenciamento de código aberto do Dlib permite que você o use em qualquer aplicativo, gratuitamente. (DAVIS E. King, 2022)

2.4. EMBEDDINGS

Um *embedding* é simplesmente uma grande lista de números, tipicamente representada como um vetor de 300 a 4096 números entre -2 e 2. Essa lista de números representa a interpretação complexa de um modelo de aprendizado de máquina do significado na entrada. As entradas podem ser texto, imagens, áudio, fotos e muito mais. É uma destilação do significado extraído o próprio modelo pode ter dezenas de milhares ou milhões de características e categorias então cada valor no embedding tem muitas informações compactadas nele. Essa densidade de informações não é algo que um humano pode seguir e, por isso, os embeddings são frequentemente considerados sem sentido se expostos.

Os embeddings são então usados matematicamente para encontrar significados semelhantes em dados armazenados, agrupar dados, encontrar anomalias e assim por diante. Uma vez que um embedding foi criado para um rosto, ele pode ser comparado a outros embeddings que foram gerados para outros rostos. Aqueles que estão “próximos” um do outro (matematicamente) são rostos similares e aqueles mais distantes são rostos diferentes. Essa técnica permite procurar alguém usando uma imagem do rosto dela, vinculando-a a informações sobre essa pessoa e usando isso para algumas das aplicações mais emocionantes (e assustadoras) de reconhecimento facial. (MURPHY Murphy, 2024)

2.5. FACE RECOGNITION 1.3.0

Face_recognition é a biblioteca de reconhecimento facial mais simples do mundo. Construída usando o reconhecimento facial de última geração do dlib, treinada utilizando aprendizado profundo. O modelo tem uma precisão de 99,3% na referência de rostos

rotulados na natureza. (GEITGEY A., 2017). No **Face_recognition**, o termo "*embeddings*" é renomeado para *encodings*, que representam uma sequência de 128 medições que descrevem os traços faciais únicos de uma pessoa. O funcionamento da biblioteca segue o seguinte fluxo:

O fluxo começa com a função `face_locations`, que identifica as regiões de uma imagem onde há rostos. Por padrão, ela utiliza o algoritmo *Histogram of Oriented Gradients* (HOG), ideal para processamento rápido em CPUs. No entanto, também é possível usar uma rede neural convolucional (CNN) para maior precisão em sistemas com GPUs, embora isso exija mais recursos computacionais.

Após localizar os rostos, a função `face_encodings` gera vetores numéricos (*encodings*) para cada rosto identificado. Esses *encodings* representam as características faciais principais de uma pessoa, e são criados usando o modelo `ResNet_model_v1`, uma rede neural convolucional que oferece uma codificação precisa e robusta, permitindo a comparação de rostos com alta acurácia.

A última etapa do fluxo é a comparação dos *encodings* gerados pela função `face_distances`. Ela utiliza a função `linalg.norm`, do NumPy, para calcular a distância euclidiana entre os *encodings* dos rostos detectados e os *encodings* conhecidos. Quanto menor a distância, maior a similaridade entre os rostos, permitindo identificar com precisão se o rosto em uma nova imagem corresponde a um já registrado.

2.6. CSV

O arquivo CSV (valores separados por vírgulas) é um arquivo de texto com formato específico para possibilitar o salvamento dos dados em um formato estruturado de tabela.

2.7. PROCESSO DE ORGANIZAÇÃO

2.7.1. Git e GitHub

Git é um sistema de controle de versão distribuído amplamente utilizado para rastrear e gerenciar alterações no código-fonte, GitHub é uma plataforma de hospedagem de código que permite hospedar repositórios Git e colaborar com outros desenvolvedores (KRIGER Brunno, 2022). Na implementação de sistemas de reconhecimento facial na educação, Git e GitHub são usados para controlar as diferentes versões do código-fonte, facilitar a colaboração entre membros da equipe de desenvolvimento e fornecer um histórico completo de alterações no projeto.

2.7.2. Figma

De acordo com seu próprio site, o Figma é uma ferramenta de design que combina a acessibilidade da web com as funcionalidades de um aplicativo nativo. Isso significa que o Figma é disponibilizado via web, com acesso através de qualquer navegador, sem a necessidade de download ou instalação de algum software. E traz todas as funcionalidades de um aplicativo desenvolvido de acordo com a base do sistema operacional em que irá operar.(LOPES Michele , 2023). Com essa ferramenta, designers e demais profissionais de todo o mundo têm a chance de construir o design de produtos digitais inteiros, como sites e aplicativos para dispositivos móveis (VILLAIN, Mateus; ISABELLE, Maria, 2022).

2.8. DOCKER

O Docker é uma plataforma de código aberto que permite aos desenvolvedores construir, implementar, executar, atualizar e gerenciar contêineres. Contêineres são componentes executáveis padronizados que combinam o código fonte da aplicação com as bibliotecas e dependências do sistema operacional necessárias para executar esse código em qualquer ambiente. (SUSNJARA, Stephanie ; SMALLEY, Ian, 2024)

2.9. PRIVACIDADE E ÉTICA SOB AS LGPD

A Lei Geral de Proteção de Dados (LGPD) é uma legislação que visa proteger a privacidade e garantir o tratamento adequado de dados pessoais. Ela impõe sanções para o descumprimento das normas, como multas e bloqueio de dados. A LGPD exige transparência no tratamento de dados sensíveis, como informações biométricas, e requer medidas de segurança robustas para garantir a conformidade (Lei nº 13.709/2018). Além da conformidade legal, é essencial que as instituições que utilizam reconhecimento facial estabeleçam protocolos claros de consentimento informado e garantam que os dados coletados sejam tratados de maneira ética e transparente. Questões como o consentimento 22 para o uso das imagens, a proteção contra possíveis violações de segurança e o direito de revisão ou exclusão dos dados são fundamentais para a implementação responsável dessa tecnologia.

O uso do reconhecimento facial levanta questões complexas relacionadas à ética e privacidade, especialmente diante de regulamentações rigorosas como a LGPD. A coleta de dados biométricos, incluindo informações faciais, requer não apenas medidas de segurança robustas, mas também adesão estrita aos princípios éticos e diretrizes legais estabelecidas pela legislação. É fundamental uma abordagem cuidadosa para assegurar a conformidade e

proteção adequadas dos dados dos usuários, considerando os impactos da LGPD no tratamento dessas informações sensíveis. Nesse sentido, a aplicação de técnicas de criptografia pode reforçar a segurança dos dados, minimizando os riscos associados ao vazamento ou uso indevido das informações coletadas.

3.0. METODOLOGIA

Este capítulo detalha o processo de desenvolvimento da aplicação. Inicialmente, as restrições do sistema foram definidas em colaboração com o Prof. Dr. Katyusco de Farias Santos, que atuou como cliente durante todo o ciclo de desenvolvimento. A equipe de desenvolvimento foi dividida em duas áreas principais: o aluno João Pedro de Sousa Corrêa ficou responsável pelo desenvolvimento do front-end, enquanto o aluno Isaac Antônio S. Barbosa assumiu a responsabilidade pelo desenvolvimento do back-end do sistema.

O sistema foi desenvolvido em duas versões distintas. A versão piloto 1.0 utiliza reconhecimento facial baseado em vídeo. No entanto, após a realização de testes, os resultados obtidos não atenderam às expectativas de precisão e desempenho. Como resultado, optou-se por modificar a abordagem, substituindo o uso de vídeos por fotos. Essa alteração teve como objetivo melhorar a precisão do reconhecimento facial e aumentar a eficiência do sistema.

3.1. RESTRIÇÕES DO SISTEMA:

Após a comunicação com o cliente foram elaborados os seguintes pontos:

- **Dispositivo móvel:** A aplicação deve ser otimizada para uso em dispositivos móveis, permitindo que o professor utilize o dispositivo para capturar imagens, tanto dos alunos individualmente quanto da turma inteira, diretamente em sala de aula, facilitando assim a coleta de dados de presença de forma prática.
- **Compatibilidade Multiplataforma:** O sistema deve ser desenvolvido como uma aplicação web, garantindo sua acessibilidade em diferentes navegadores e plataformas.
- **Geração de Lista de Presença:** Ao concluir o processo de reconhecimento facial, o sistema deve gerar uma lista de alunos presentes e ausentes, permitindo que o professor realize uma verificação final e confirme a chamada.
- **Gestão de Múltiplas Turmas:** O sistema deve permitir que o professor cadastrado crie e gerencie várias turmas vinculadas à sua conta, facilitando o controle e organização das atividades acadêmicas.

Com base nas especificações iniciais fornecidas pelo cliente, foram definidos os requisitos funcionais e não funcionais do sistema. Esses requisitos são essenciais para garantir o funcionamento completo do sistema, desde a captura das imagens até a geração dos relatórios de presença.

3.1.1. Requisitos funcionais

Requisitos funcionais definem as ações e comportamentos que o sistema deve realizar para atender às necessidades dos usuários, especificando o que o sistema deve fazer.

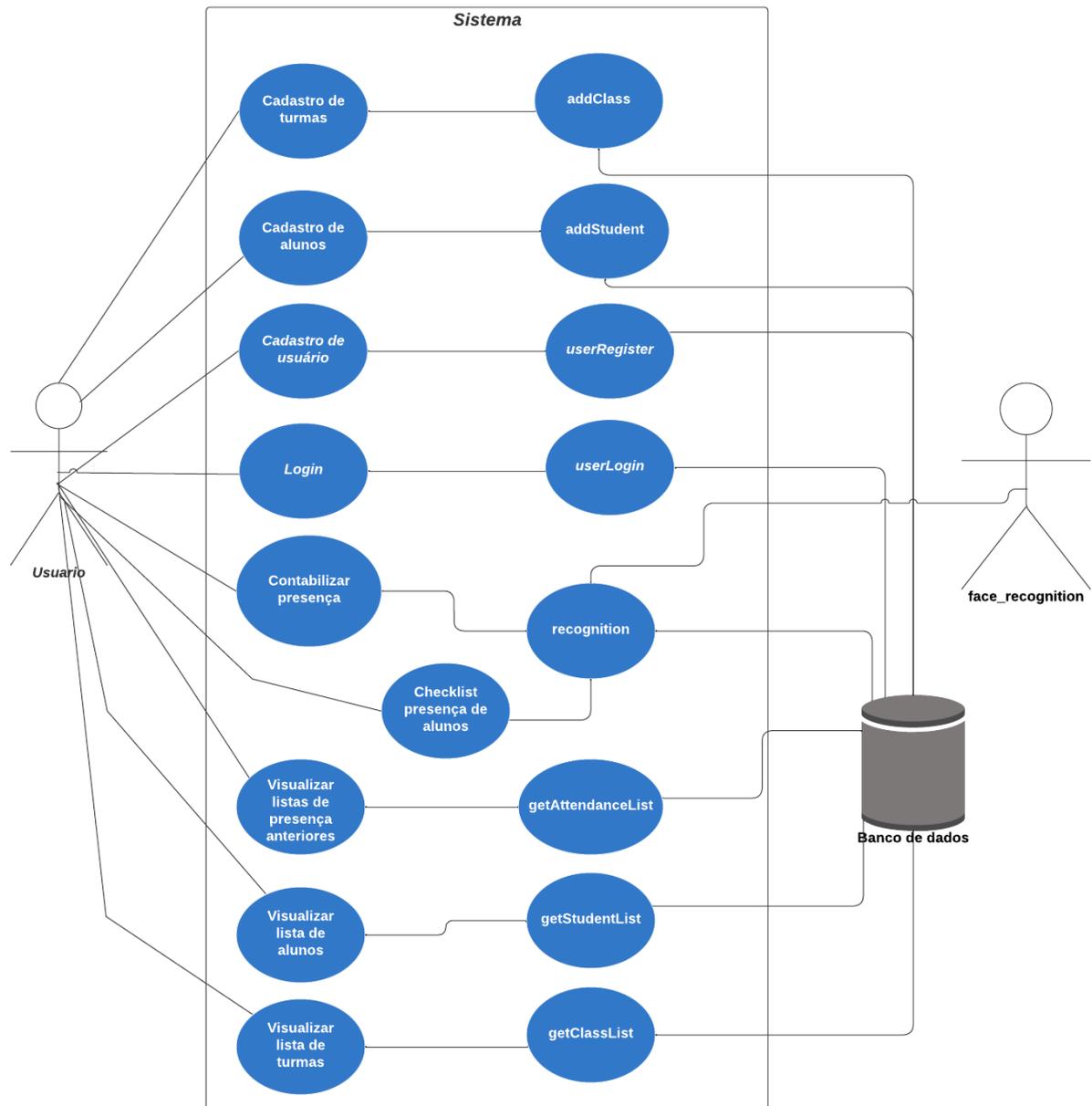
- **Cadastro de usuário:** O sistema deve permitir o cadastro de novos usuários, como professores, com informações essenciais, incluindo nome, e-mail e senha, garantindo a segurança e a individualização do acesso ao sistema.
- **Login:** O sistema deve fornecer uma funcionalidade de login segura, permitindo que os usuários registrados acessem suas contas com autenticação baseada em e-mail e senha.
- **Cadastro de Alunos:** O sistema deve permitir que os professores cadastrem os alunos, incluindo informações como nome, número de matrícula e foto para o reconhecimento facial.
- **Cadastro de Turmas:** O sistema deve permitir que os professores criem turmas e associam alunos a essas turmas, facilitando a organização e o gerenciamento das listas de presença.
- **Gerar Lista de Presença:** Após o processo de reconhecimento facial, o sistema deve gerar automaticamente uma lista de alunos presentes e ausentes. Essa lista deve permitir que o professor verifique sua precisão e faça correções, caso haja algum erro no reconhecimento.
- **Visualizar Listas de Presença:** O sistema deve permitir que os professores acessem as listas de presença geradas em datas anteriores, com a opção de filtrar por data e nome da turma. Ao selecionar uma turma específica, será exibido um visualizador de CSV integrado, com a possibilidade de baixar o arquivo diretamente.
- **Contabilizar Presença:** O sistema deve calcular automaticamente as presenças dos alunos com base nas imagens capturadas da turma, associando cada rosto identificado ao registro correspondente no banco de dados.
- **Campos de Busca:** O sistema deve incluir campos de busca que permitam aos professores localizar rapidamente alunos, turmas e listas de presença, melhorando a eficiência da navegação e do gerenciamento de dados.

3.1.2. Requisitos não funcionais:

- **Persistência de dados:** O sistema deve garantir a persistência e integridade dos dados armazenados, utilizando um banco de dados confiável que suporte operações transacionais.
- **Portabilidade:** A aplicação deve ser compatível com diferentes dispositivos e sistemas operacionais, incluindo smartphones, tablets e computadores, assegurando que o sistema funcione de maneira consistente em navegadores modernos.
- **Performance:** O sistema deve ser capaz de processar dados de forma eficiente, especialmente durante o reconhecimento facial e a geração de listas de presença, assegurando uma resposta rápida. O tempo máximo de processamento para essas operações não deve exceder 30 segundos, garantindo uma experiência de usuário fluida e sem atrasos perceptíveis.
- **Conformidade:** O sistema deve estar em conformidade com as leis e regulamentações aplicáveis, como a Lei Geral de Proteção de Dados (LGPD), garantindo que todos os dados pessoais sejam tratados de maneira segura e em conformidade com as diretrizes de privacidade.
- **Segurança:** O sistema deve implementar medidas de segurança, como criptografia de dados sensíveis, autenticação segura e proteção contra acesso não autorizado, garantindo a confidencialidade e integridade dos dados.

Com base nos requisitos definidos, foi elaborado um diagrama de caso de uso para ilustrar as interações entre os usuários e as funcionalidades do sistema. A Figura 1 apresenta esse diagrama, demonstrando as operações que o sistema deve suportar, bem como os atores envolvidos em cada processo.

Figura 1 – Diagrama de casos de uso do sistema



Fonte: autoria própria

3.2. PESQUISA BIBLIOGRÁFICA

Após a definição dos requisitos do sistema, foi realizada uma pesquisa bibliográfica com o objetivo de identificar tecnologias já existentes que pudessem ser aplicadas ou adaptadas ao desenvolvimento da aplicação. Esta pesquisa concentrou-se em duas áreas principais: o uso de soluções de reconhecimento facial e as tecnologias de desenvolvimento web que suportam a criação de sistemas multiplataforma.

A pesquisa bibliográfica envolveu a revisão superficial de artigos, documentações em fontes online que discutem o uso de reconhecimento facial em diversos contextos. Além disso, foram investigadas as melhores práticas e ferramentas de desenvolvimento web, com foco em frameworks e bibliotecas que garantam eficiência, segurança e escalabilidade ao sistema proposto.

Os resultados obtidos nesta pesquisa permitiram a seleção de tecnologias e abordagens que atendem aos requisitos do projeto e enfrentam os desafios comuns em projetos semelhantes. Com base nesses resultados, as seguintes tecnologias foram escolhidas para a implementação do sistema:

3.2.1 Front-end

- **React (v 18.2.0):** React é um framework JavaScript. O React foi originalmente criado por engenheiros do Facebook para resolver os desafios envolvidos no desenvolvimento de interfaces de usuário complexas com conjuntos de dados que mudam com o tempo (GACKENHEIMER Cory, 2015). Ele permite a criação de componentes reutilizáveis que facilitam o desenvolvimento de aplicações dinâmicas e interativas. A escolha do React se deve à sua popularidade, desempenho e grande comunidade de suporte, o que o torna ideal para projetos de médio a grande porte como o nosso.
- **Next.js (v 13.2.4):** Next.js é um framework React para construir aplicativos web full-stack. Você usa React Components para construir interfaces de usuário, e Next.js para recursos e otimizações adicionais. Por baixo dos panos, o Next.js também abstrai e configura automaticamente as ferramentas necessárias para o React, como empacotamento, compilação e muito mais (Vercel, 2024). A combinação de React com Next.js foi escolhida para garantir que o sistema tenha uma estrutura robusta, com bom desempenho e excelente experiência do usuário.
- **Axios (v 1.4.0):** Axios é um cliente HTTP baseado em Promises para fazer requisições. Poderá ser utilizado tanto no navegador quanto no Node.js ou qualquer serviço de API. (SOUSA Rômulo, 2021). É uma solução flexível e simples para a comunicação entre o front-end e o back-end, permitindo a integração fácil com APIs. Foi escolhido o Axios por sua capacidade de gerenciar requisições de forma assíncrona, o que facilita a implementação de fluxos de autenticação e tratamento de erros.

- **Material-UI (v 5.12.5):** Material UI é uma biblioteca de componentes React de código aberto que implementa o Material Design do Google. Ele inclui uma coleção abrangente de componentes pré-fabricados prontos para uso na produção e apresenta um conjunto de opções de personalização que facilitam a implementação do seu próprio sistema de design personalizado sobre nossos componentes (MUI, 2024). O uso de Material-UI no projeto foi essencial para acelerar o desenvolvimento da interface e garantir uma aparência profissional e intuitiva.
- **Next-auth (v 4.22.1):** Next-auth é uma solução de autenticação completa para aplicações Next.js (Newark Codie, 2024). Ela suporta várias estratégias de autenticação, entre elas a autenticação baseada em e-mail/senha. Foi utilizado Next-auth para gerenciar a autenticação dos usuários no sistema de forma segura e integrada, facilitando a gestão de sessões.
- **React-dropzone (v 14.2.3):** React-dropzone é uma biblioteca React para criar zonas de arrastar e soltar arquivos, permitindo uma experiência de upload mais intuitiva e moderna. Foi utilizada no projeto para facilitar o upload de imagens, como as fotos dos alunos, garantindo uma interface simples e intuitiva para o usuário.
- **React-hook-form (v 7.44.3):** React-hook-form é uma biblioteca para gerenciamento de formulários em React, que se destaca pela simplicidade e eficiência no manuseio de inputs e validações. Foi adotada no projeto para gerenciar os formulários de cadastro de usuários, alunos e turmas, proporcionando uma forma mais performática e menos verbosa de lidar com dados de formulários no React.
- **Compressorjs (v 1.2.1):** Compressor de imagem JavaScript. Usa a API nativa canvas.toBlob do navegador para fazer o trabalho de compressão, o que significa que é uma compressão com perdas, assíncrona e tem efeitos de compressão diferentes em navegadores diferentes. Geralmente, use isso para pré-comprimir uma imagem no lado do cliente antes de carregá-la.

3.2.2. Back-end

- **Python (v 3.8.10):** O Python é uma linguagem de programação interpretada, interativa e orientada a objetos. O mesmo incorporou módulos, exceções, tipagem dinâmica, tipos de dados dinâmicos de alto nível e classes. Há suporte a vários paradigmas de programação além da programação orientada a objetos, tal como programação procedural e funcional (PythonSoftware Foundation, 2024). No projeto, Python foi

escolhido para o desenvolvimento do back-end devido à sua robustez, grande biblioteca de recursos e suporte a frameworks poderosos como Django, que facilitam a criação de aplicações web seguras e escaláveis.

- **Django (v 4.2):** O Django é um framework web Python de código aberto, que se destaca por oferecer um ambiente que simplifica a criação de soluções web escaláveis, ao mesmo tempo, em que promove o desenvolvimento rápido e um design limpo, proporcionando ferramentas robustas e eficientes para pessoas desenvolvedoras (URANO Lais, 2023). A escolha de Django foi motivada pela sua maturidade, segurança embutida, e ampla documentação.
- **Face recognition (v 1.3.0):** `face_recognition` é uma biblioteca Python amplamente usada para reconhecimento facial. Ela é construída sobre `dlib`, uma poderosa biblioteca de aprendizado de máquina, e oferece uma interface simples para detecção e reconhecimento de rostos em imagens e vídeos. No projeto, `face_recognition` foi utilizada para processar as imagens capturadas e identificar os rostos dos alunos, comparando-os com os registros existentes para contabilizar a presença. Sua implementação simplificou significativamente a adição de funcionalidades de reconhecimento facial ao sistema.
- **PostgreSQL (v 15):** O PostgreSQL é uma ferramenta que atua como sistema de gerenciamento de bancos de dados relacionados. Seu foco é permitir implementação da linguagem SQL em estruturas, garantindo um trabalho com os padrões desse tipo de ordenação dos dados (SOUZA Ivan, 2021). No projeto, PostgreSQL foi escolhido como o banco de dados principal para armazenar informações críticas, como dados dos usuários, turmas, imagens e encodings faciais. Sua capacidade de lidar com grandes volumes de dados e operações complexas, aliada à fácil integração com Django, fez do PostgreSQL a escolha ideal para garantir a escalabilidade e segurança do sistema.

Essa combinação de tecnologias, baseada nas pesquisas, assegura que o sistema desenvolvido seja capaz de atender às necessidades do usuário.

3.3. DESENVOLVIMENTO DA VERSÃO PILOTO 1.0 DO SISTEMA

O desenvolvimento da versão piloto 1.0 do sistema iniciou-se com a criação de um protótipo de interface gráfica no Figma. Este protótipo desempenhou um papel crucial como referência visual, orientando o desenvolvimento do *front-end*. Simultaneamente, iniciou-se o

desenvolvimento do *back-end*, assegurando que as funcionalidades do sistema fossem implementadas de forma paralela e integrada. Esse trabalho conjunto entre o *front-end* e o *back-end* garantiu que ambas as partes do sistema evoluíssem de maneira harmoniosa, resultando em uma integração eficiente entre as funcionalidades de reconhecimento facial e a interface de usuário.

3.3.1. Desenvolvimento do *front-end*

Baseado no protótipo de interface gráfica criado no Figma, as telas desenvolvidas foram desenhadas com base nos requisitos funcionais do sistema e projetadas para serem intuitivas e fáceis de usar.

O foco principal no design da interface de usuário foi garantir uma experiência fluida, permitindo que os professores utilizassem o sistema sem dificuldades técnicas, mesmo em situações de uso intensivo, como durante a chamada em sala de aula. Cada elemento da interface foi pensado para facilitar a navegação e a execução das tarefas necessárias, desde o cadastro de alunos até a visualização e correção das listas de presença geradas.

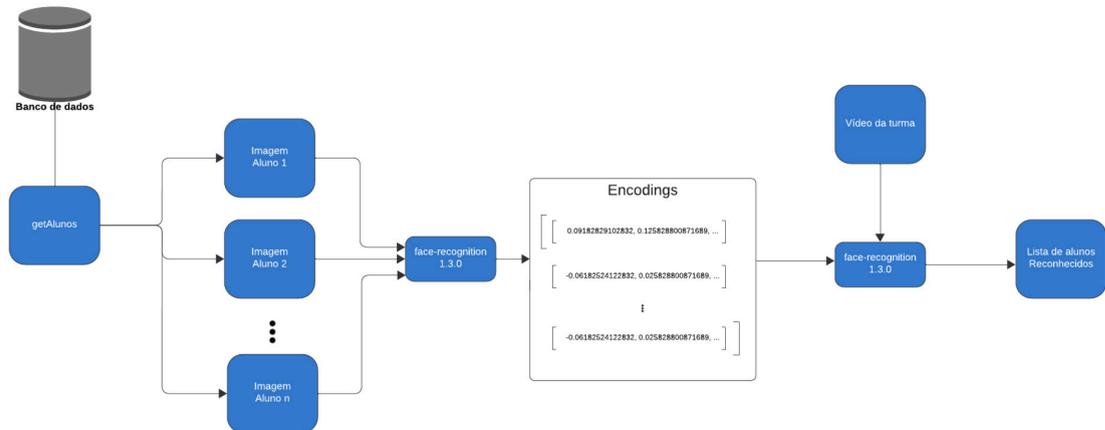
Para garantir uma comunicação eficiente entre o *front-end* e o *back-end*, foi utilizada a biblioteca Axios. Com isso, o *front-end* é capaz de enviar e receber dados de forma rápida e eficiente, como no caso de operações de login, cadastro de alunos, e recuperação de listas de presença. Essa integração foi essencial para manter a aplicação responsiva e garantir que as atualizações no *front-end* reflitam imediatamente as operações realizadas no *back-end*, proporcionando uma experiência de usuário contínua e sem interrupções.

3.3.2. Desenvolvimento do *back-end*

O desenvolvimento do *back-end* focou na criação de uma arquitetura escalável para suportar as funcionalidades principais do sistema. Utilizando Python e o *framework* Django, foi iniciada a implementação dos modelos de banco de dados, que foram projetados para gerenciar eficientemente as informações de usuários, alunos, turmas, e os dados de presença.

Após a modelagem, foi desenvolvido o CRUD (*Create, Read, Update, Delete*) de usuários, turmas e alunos. Em seguida foi desenvolvida a função *recognition* que será utilizada para a contabilização de presença de alunos. A seguir na Figura 2 é demonstrado o fluxo da função *recognition*.

Figura 2 – Fluxo da função *recognition*



Fonte: autoria própria

A função *recognition* realiza primeiramente uma consulta ao banco de dados para obter os alunos associados à turma selecionada. Para cada aluno, recupera-se a imagem cadastrada, e em seguida é gerado um encoding dessa imagem utilizando a função *face_encodings* da biblioteca *face_recognition*. Esses *encodings* são armazenados em um mapa, associando cada encoding ao respectivo aluno.

Com o mapa de *encodings* dos alunos pronto, o próximo passo é processar o vídeo da turma. A cada *frame* capturado, utiliza-se a função *face_locations*, também da biblioteca *face_recognition*, para identificar as regiões no *frame* que contém rostos. Após a localização, a função *face_encodings* é novamente aplicada nas regiões detectadas para gerar uma lista de *encodings* referentes aos rostos presentes naquele *frame*.

Por fim, a comparação entre os encodings dos alunos e os encodings gerados a partir do vídeo é realizada utilizando a função *face_distances*, da mesma biblioteca. Essa função calcula a distância entre cada par de *encodings*, permitindo determinar a correspondência entre os rostos no vídeo e os alunos cadastrados, viabilizando o reconhecimento dos presentes.

Em seguida foi desenvolvida a API que conecta o *back-end* com o *front-end*. Essa API permite a comunicação entre as interfaces, facilitando operações como o cadastro de novos alunos, a criação de turmas e o envio de dados para o processo de reconhecimento facial. A API foi projetada com *endpoints* específicos para cada funcionalidade, garantindo a modularidade e a facilidade de manutenção.

Além disso, foi implementada medidas de segurança, como autenticação segura e controle de acesso, assegurando que os dados dos alunos e das listas de presença sejam tratados com confidencialidade e integridade.

Após a conclusão do desenvolvimento do *front-end* e do *back-end*, a versão piloto 1.0 completa do sistema foi criada, utilizando vídeos previamente gravados para realizar o reconhecimento facial. Essa versão inicial integrou todas as funcionalidades principais, desde a captura e processamento das imagens até a geração e visualização das listas de presença, permitindo que o sistema fosse testado e validado em um ambiente controlado.

3.4. DEMONSTRAÇÃO DA VERSÃO PILOTO 1.0 DO SISTEMA

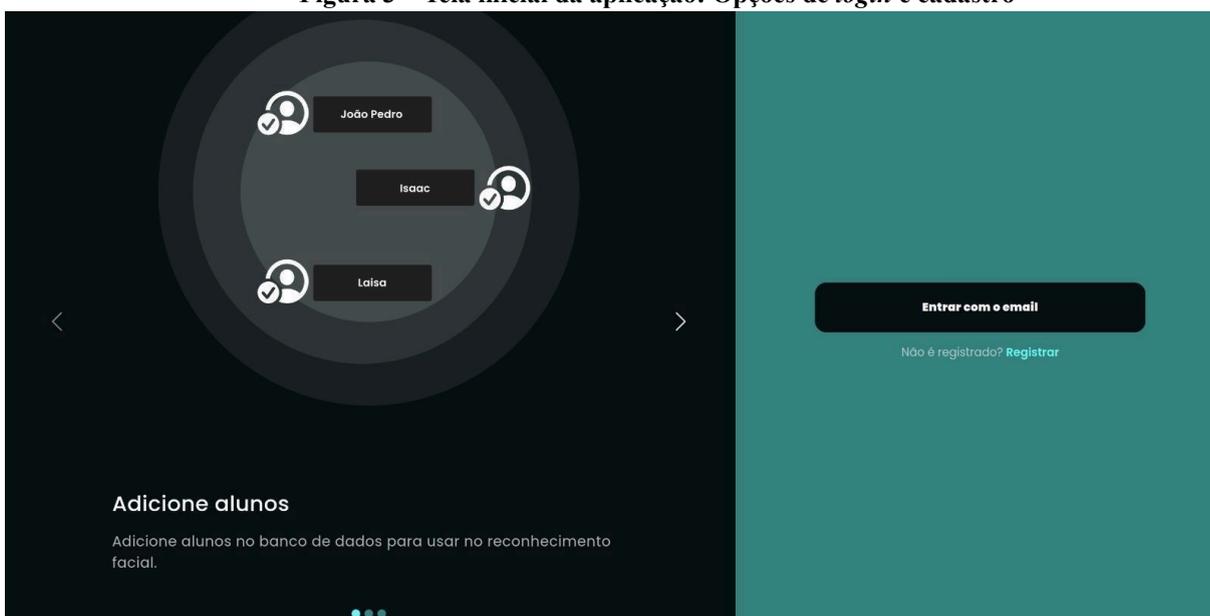
Nesta seção, são exibidas algumas ilustrações da aplicação com base na versão piloto 1.0 do sistema.

3.4.1. Login e Cadastro

A seguir, apresentamos os processos de *login* e cadastro da aplicação. As imagens ilustram as telas correspondentes a essas funcionalidades:

- Tela Inicial: Esta é a primeira tela apresentada ao usuário ao iniciar a aplicação. Ela oferece opções para o usuário escolher entre acessar a aplicação por meio do login ou criar uma nova conta através do cadastro. O design é intuitivo e permite uma navegação fácil entre as opções disponíveis.

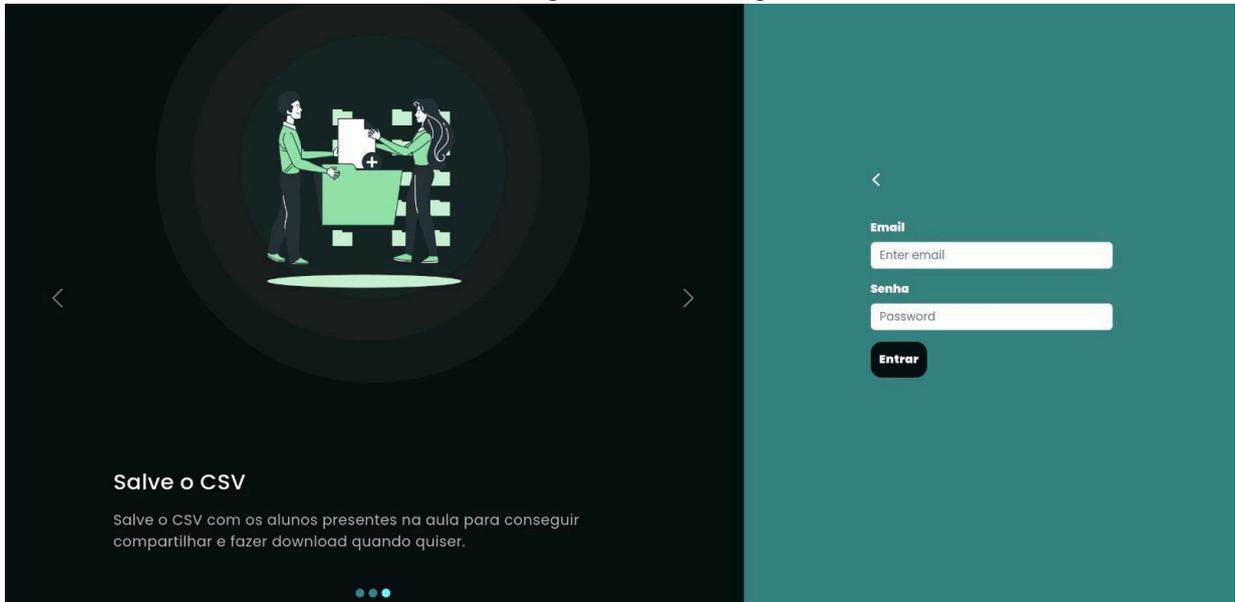
Figura 3 – Tela inicial da aplicação: Opções de *login* e cadastro



Fonte: autoria própria

- Tela de *Login*: Esta tela permite que usuários existentes ingressem no sistema usando suas credenciais. O *design* é simples e direto, com campos para inserir o email do usuário e a senha, além de um botão para acessar a aplicação.

Figura 4 – Tela de login



Fonte: autoria própria

- Tela de cadastro: Nesta tela, novos usuários podem criar uma conta na aplicação. O formulário de cadastro solicita informações essenciais, como nome, endereço de e-mail e senha. Para finalizar o registro, é necessário aceitar os termos de uso. Após preencher todos os campos obrigatórios e concordar com os termos de uso, o usuário deve clicar no botão "Cadastrar" para completar o processo e acessar a aplicação.

Figura 5 – Tela de cadastro para novos usuários

Grave vídeo

Grave um vídeo da turma para marcar presença dos alunos presentes na sala de aula.

Nome
 Digite seu nome

Email
 Digite seu email

Senha
 Digite sua senha

Confirmar Senha
 Confirme sua senha

Aceito os [termos](#) de uso

Cadastrar

Fonte: autoria própria

3.4.2. Termos

A aba de Termos de Consentimento é acessada pelos usuários quando eles clicam para visualizar os termos completos relacionados ao uso do sistema. Esta tela apresenta um resumo claro e detalhado dos Termos de Uso e da Política de Privacidade, assegurando que os usuários tenham plena compreensão das condições e regras antes de prosseguir com a utilização do sistema.

Figura 6 – Tela de termos de consentimento

Termos de Uso e Política de Privacidade

1. Introdução

Este documento estabelece os Termos de Uso e a Política de Privacidade do sistema de reconhecimento facial desenvolvido para contabilizar a presença de alunos em sala de aula. Ao utilizar este sistema, você concorda com os termos aqui descritos.

2. Uso do Sistema

O sistema foi desenvolvido para uso exclusivo em ambientes educacionais para facilitar a gestão de presença de alunos em sala de aula. O acesso é restrito a usuários autorizados, como administradores escolares e professores, que utilizarão o sistema para fins educativos.

3. Coleta de Dados

O sistema coleta imagens faciais dos alunos exclusivamente para o propósito de identificação e registro de presença em sala de aula. Esses dados são capturados em tempo real e são necessários para o funcionamento correto do sistema.

4. Armazenamento e Proteção de Dados

As imagens e os dados associados são armazenados de forma segura, utilizando tecnologias de criptografia para proteger contra acessos não autorizados. O armazenamento dos dados segue as melhores práticas de segurança da informação e cumpre com as regulamentações locais, incluindo a Lei Geral de Proteção de Dados (LGPD).

5. Uso e Compartilhamento de Dados

Os dados coletados pelo sistema são utilizados apenas para a gestão de presença dos alunos e não serão compartilhados com terceiros, exceto quando exigido por lei. O sistema garante que as informações não serão utilizadas para outros fins, como marketing ou venda de dados.

6. Direitos dos Usuários

Os alunos, ou seus responsáveis, têm o direito de acessar, corrigir ou solicitar a exclusão dos seus dados pessoais armazenados pelo sistema. As solicitações devem ser enviadas ao administrador do sistema, que tomará as medidas necessárias para garantir a conformidade com a LGPD.

7. Consentimento

Ao participar das atividades em sala de aula onde o sistema é utilizado, os alunos e seus responsáveis consentem com a coleta e o processamento das imagens faciais para os fins descritos neste documento.

8. Alterações nos Termos

Este Termos de Uso e Política de Privacidade podem ser atualizados periodicamente. Os usuários serão notificados sobre quaisquer alterações significativas.

9. Contato

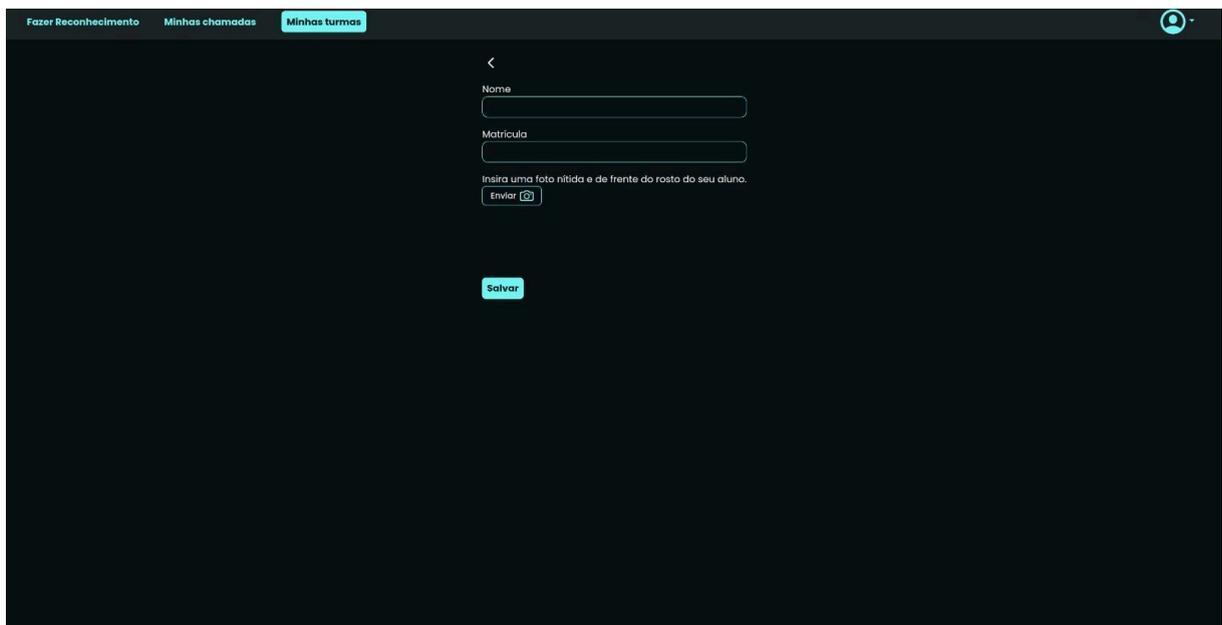
Em caso de dúvidas ou solicitações relacionadas a este Termo de Uso e Política de Privacidade, entre em contato com o administrador do sistema através dos e-mails: isaac.barbosa@academico.ifpb.edu.br e joao.correa@academico.ifpb.edu.br.

Fonte: autoria própria

3.4.3. Criação de Alunos

A tela de criação de alunos permite que novos alunos sejam cadastrados no sistema. O formulário solicita informações básicas do aluno, como nome, matrícula, e foto de identificação, que será utilizada no processo de reconhecimento facial.

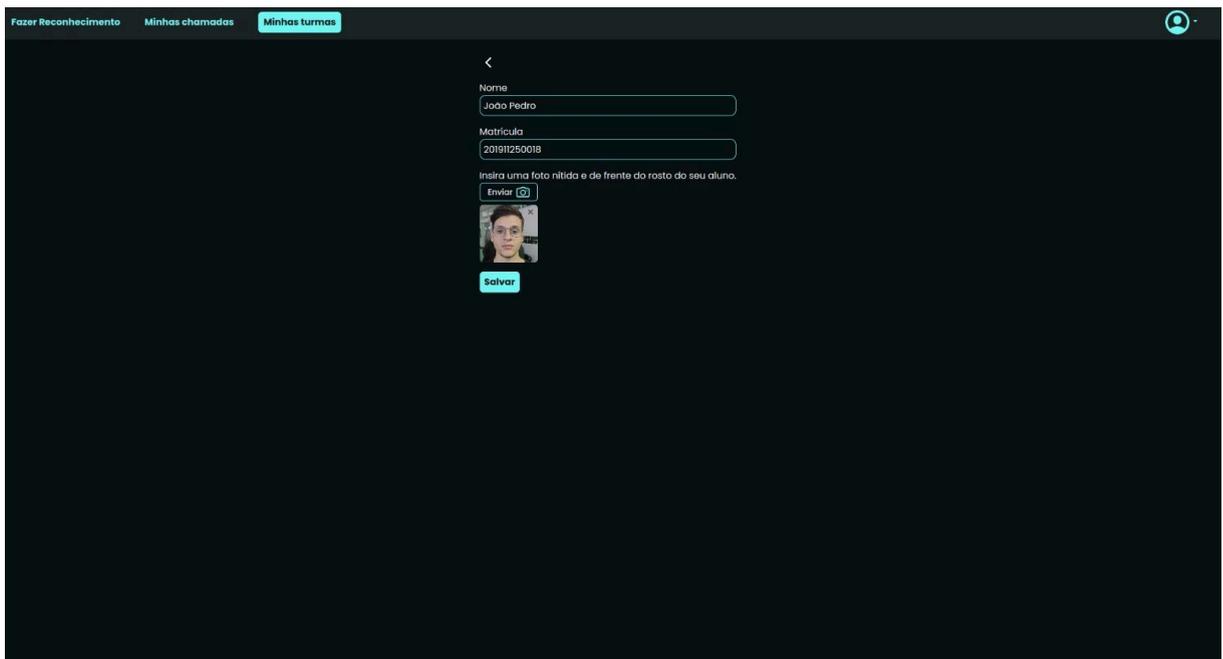
Figura 7 – Tela de criação de alunos



A imagem mostra a interface de usuário para a criação de um novo aluno. No topo, há uma barra de navegação com três opções: 'Fazer Reconhecimento', 'Minhas chamadas' e 'Minhas turmas', esta última selecionada. Abaixo, há um formulário com os seguintes elementos:

- Um ícone de seta para trás (<) no canto superior esquerdo.
- Um campo de texto rotulado 'Nome'.
- Um campo de texto rotulado 'Matrícula'.
- Uma instrução: 'Insira uma foto nítida e de frente do rosto do seu aluno.'
- Um botão rotulado 'Enviar' com um ícone de câmera.
- Um botão rotulado 'Salvar' na base do formulário.

Fonte: autoria própria

Figura 8 – Tela de criação de alunos preenchida

Fazer Reconhecimento Minhas chamadas **Minhas turmas**

<

Nome
João Pedro

Matrícula
201911250018

Insira uma foto nítida e de frente do rosto do seu aluno.

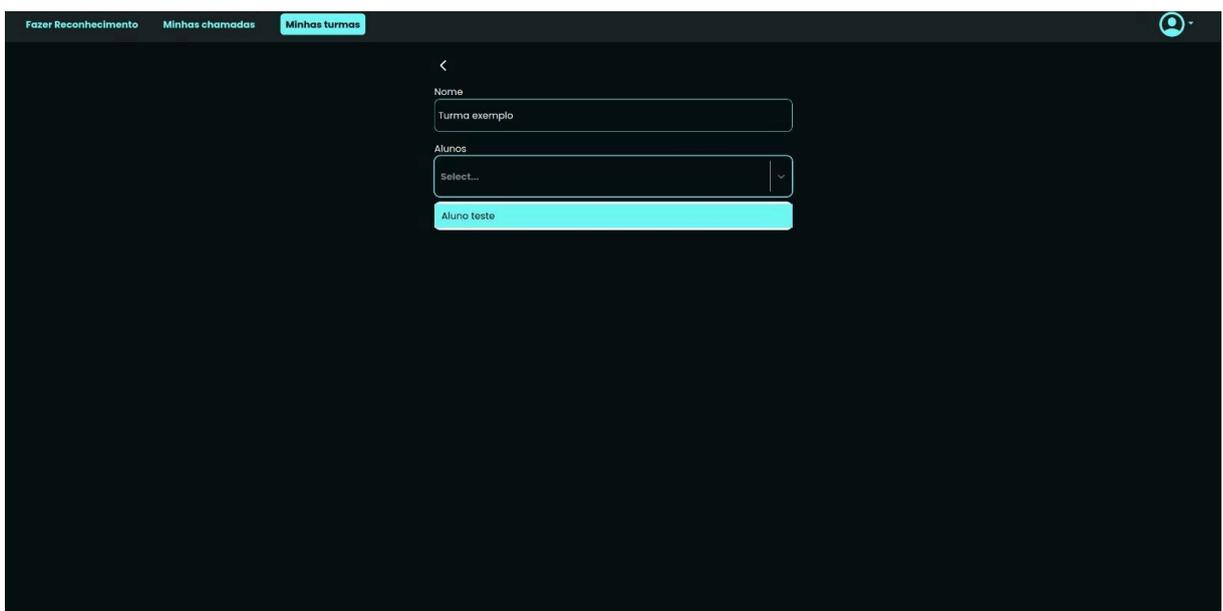
Enviar

Salvar

Fonte: autoria própria

3.4.4. Criação de Turmas

Nesta tela, o usuário pode criar novas turmas, definindo um nome para cada uma. Além disso, é possível adicionar alunos previamente cadastrados à nova turma. Ao clicar na seleção de alunos, será exibida uma lista com todos os alunos relacionados ao professor. O usuário pode então selecionar múltiplos alunos de uma vez, facilitando a inclusão dos estudantes na turma.

Figura 9 – Tela de criação de turma

Fazer Reconhecimento Minhas chamadas **Minhas turmas**

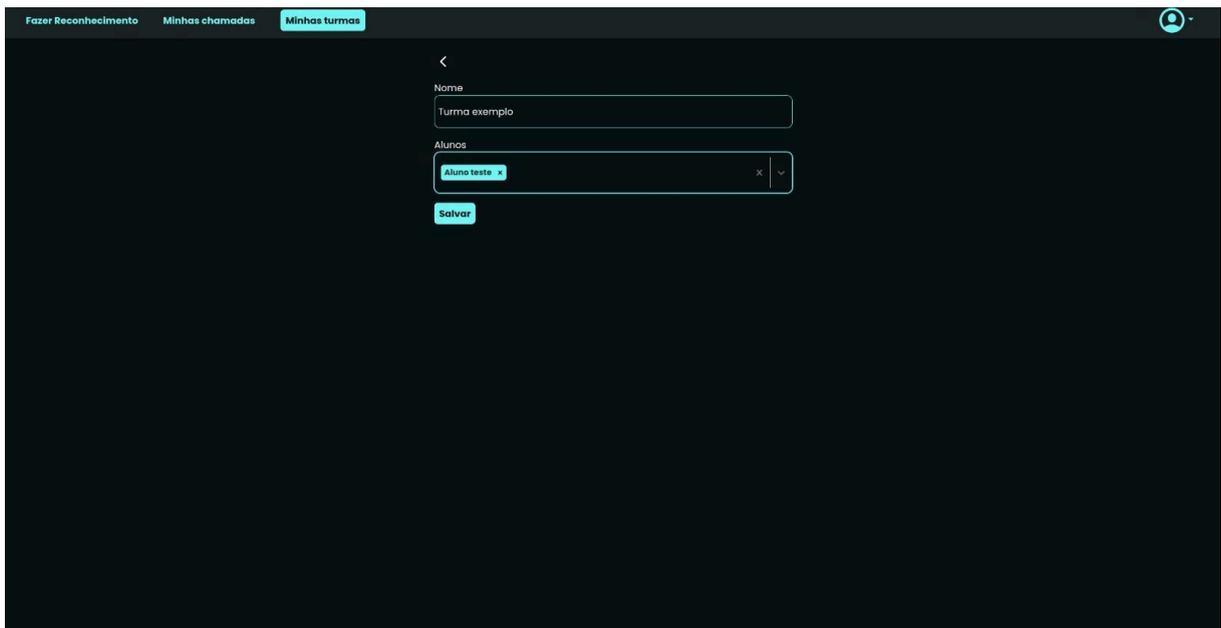
<

Nome
Turma exemplo

Alunos
Select...
Aluno teste

Fonte: autoria própria

Figura 10 – Tela de criação de turma preenchida



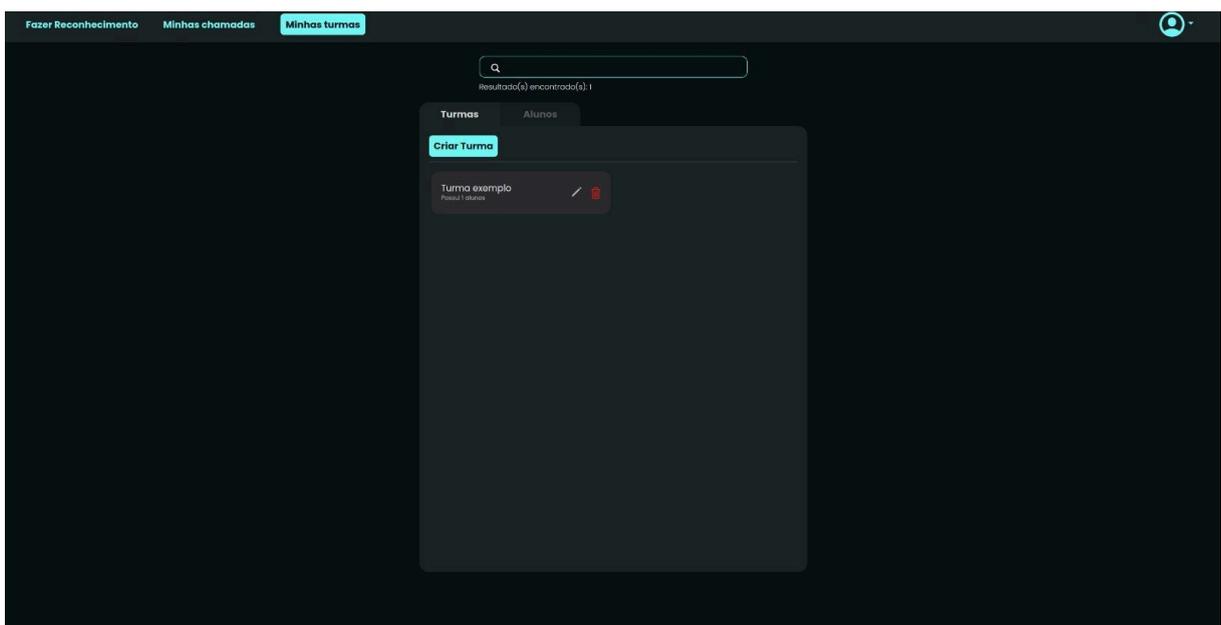
The screenshot shows a mobile application interface with a dark theme. At the top, there is a navigation bar with three tabs: 'Fazer Reconhecimento', 'Minhas chamadas', and 'Minhas turmas'. The 'Minhas turmas' tab is active. Below the navigation bar, there is a back arrow icon. The main content area contains a form with the following elements: a text input field labeled 'Nome' containing the text 'Turma exemplo'; a section labeled 'Alunos' containing a list of students, with one student 'Aluno teste' selected and marked with an 'x' and a dropdown arrow; and a 'Salvar' button at the bottom.

Fonte: autoria própria

3.4.5. Turmas e Alunos

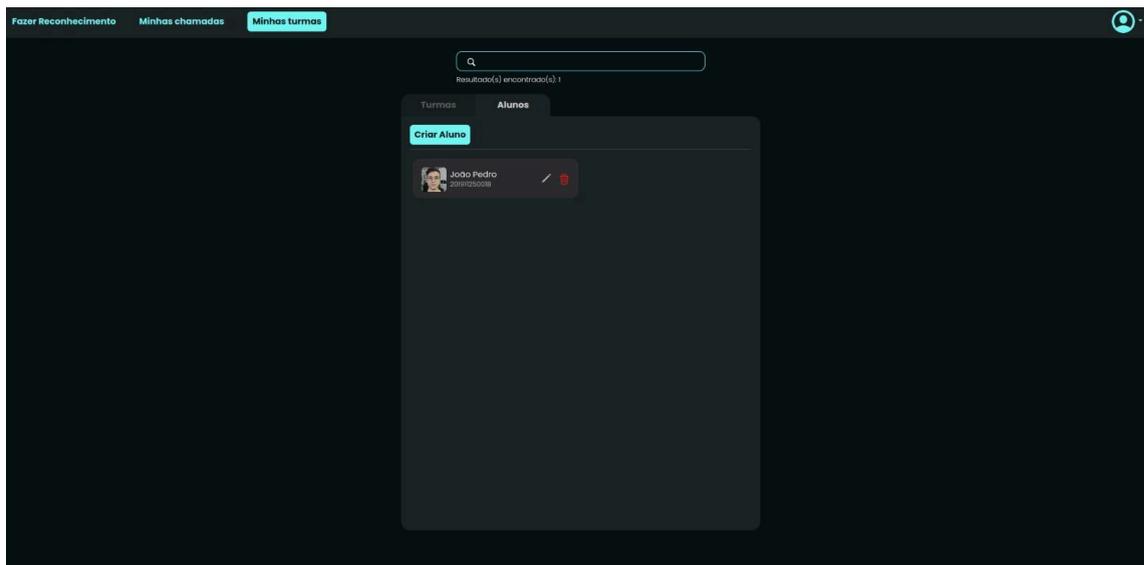
Estas telas permitem a visualização das turmas e dos alunos cadastrados no sistema. A tela de turmas exibe uma lista com o nome de cada turma e a quantidade de alunos em cada uma. Já a tela de alunos mostra uma lista com as fotos, nomes e matrículas de cada estudante, proporcionando uma visão dos alunos cadastrados. Ambas as telas facilitam a gestão e o acesso às informações básicas do sistema.

Figura 11 – Tela de visualização de turmas



Fonte: autoria própria

Figura 12 – Tela de visualização de alunos

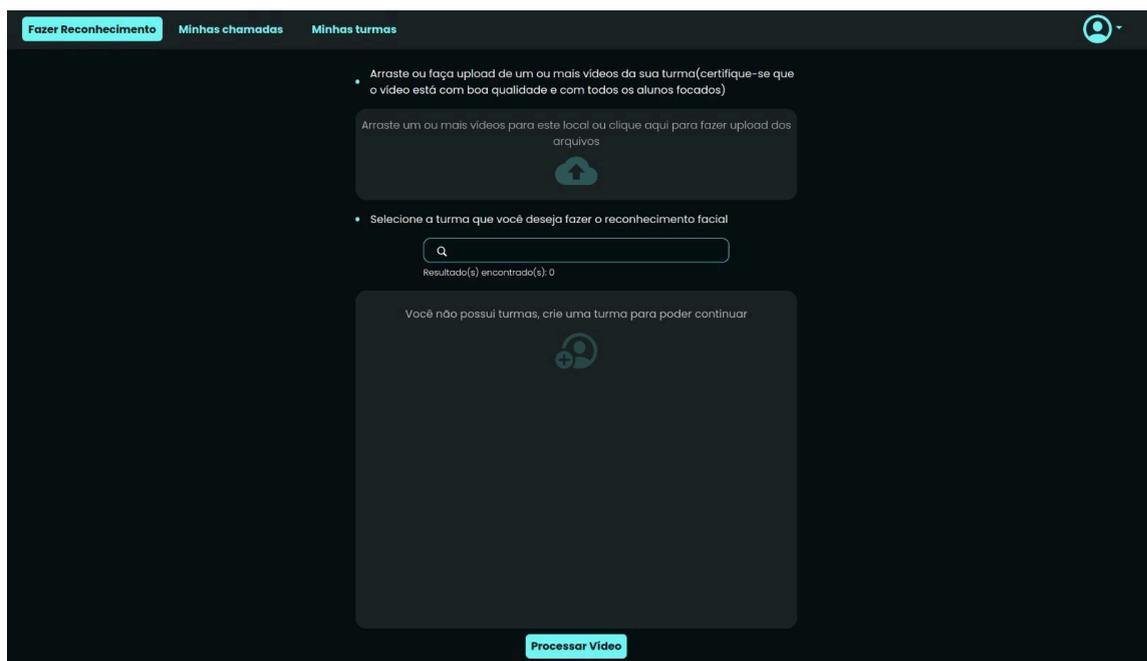


Fonte: autoria própria

3.4.6. Reconhecimento

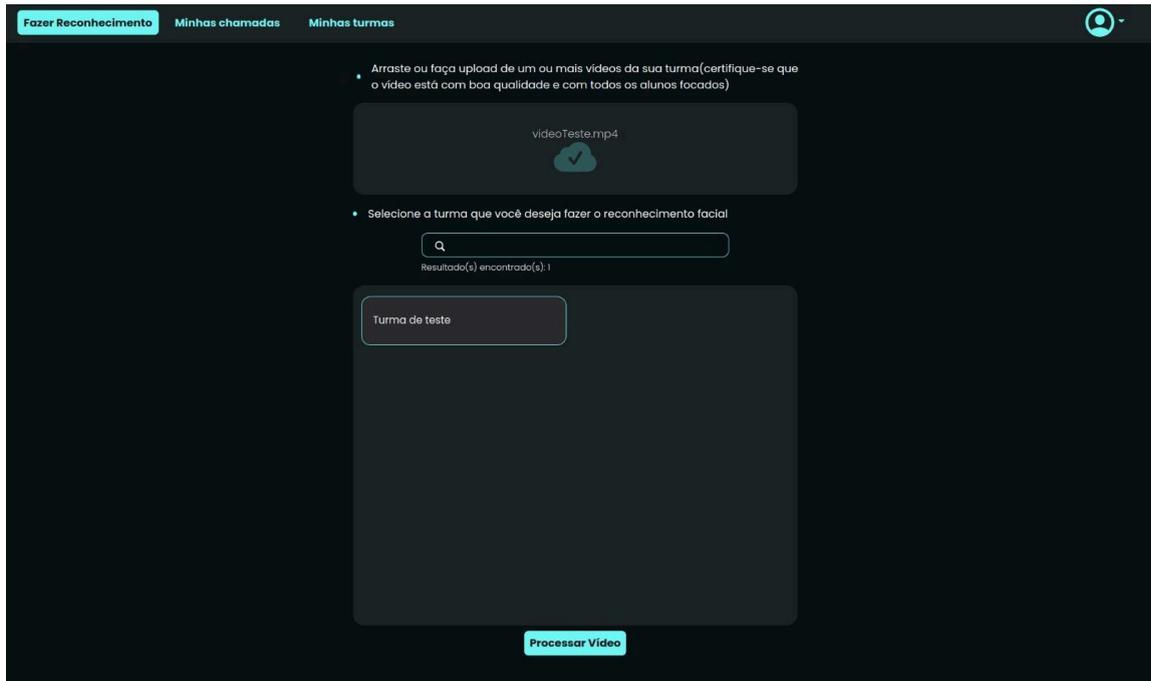
A interface principal do sistema de reconhecimento facial foi projetada para capturar e identificar os rostos dos alunos presentes na sala de aula de maneira eficiente. No campo "Upload de vídeos", o usuário pode carregar os vídeos capturados da turma durante a aula. Em seguida, no campo "Selecionar Turma", é necessário escolher a turma correspondente ao vídeo carregado. Esta combinação permite ao sistema realizar a identificação precisa dos alunos e gerar o registro de presença.

Figura 13 – Tela de reconhecimento



Fonte: autoria própria

Figura 14 – Tela de reconhecimento preenchida

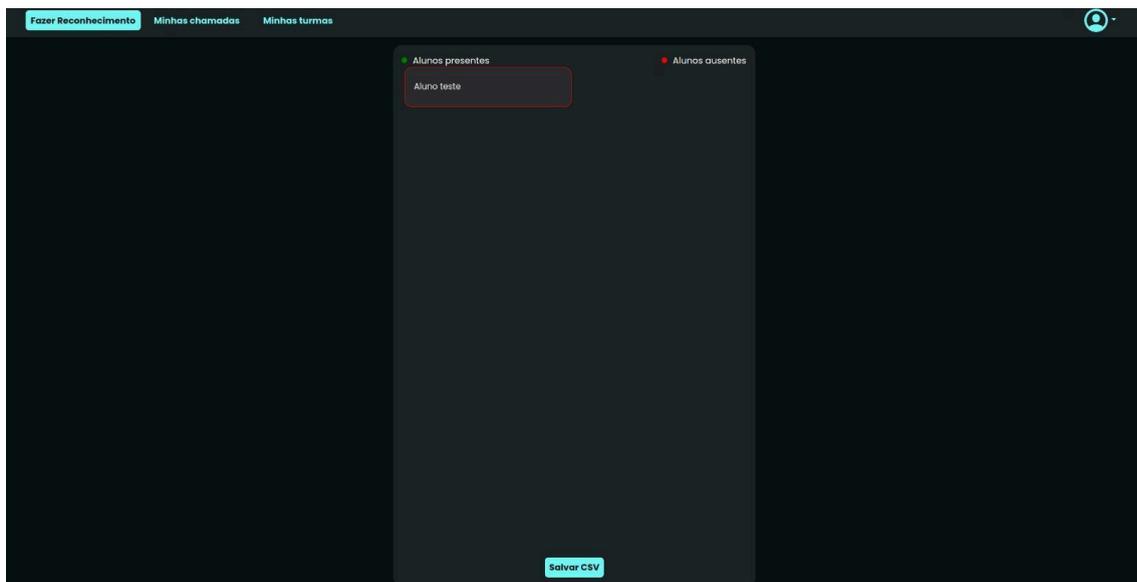


Fonte: autoria própria

3.4.7. Confirmação da Lista de Alunos Presentes

Nesta tela, o usuário pode visualizar uma lista dos alunos para os quais foi realizada a análise de reconhecimento facial. Cada nome na lista pode ser pressionado para alternar o status do aluno entre "Presente" e "Ausente". Após revisar e ajustar os status conforme necessário, o usuário deve clicar no botão "Salvar CSV" para finalizar e registrar a presença de cada aluno de forma definitiva.

Figura 15 – Tela de verificação da lista de alunos

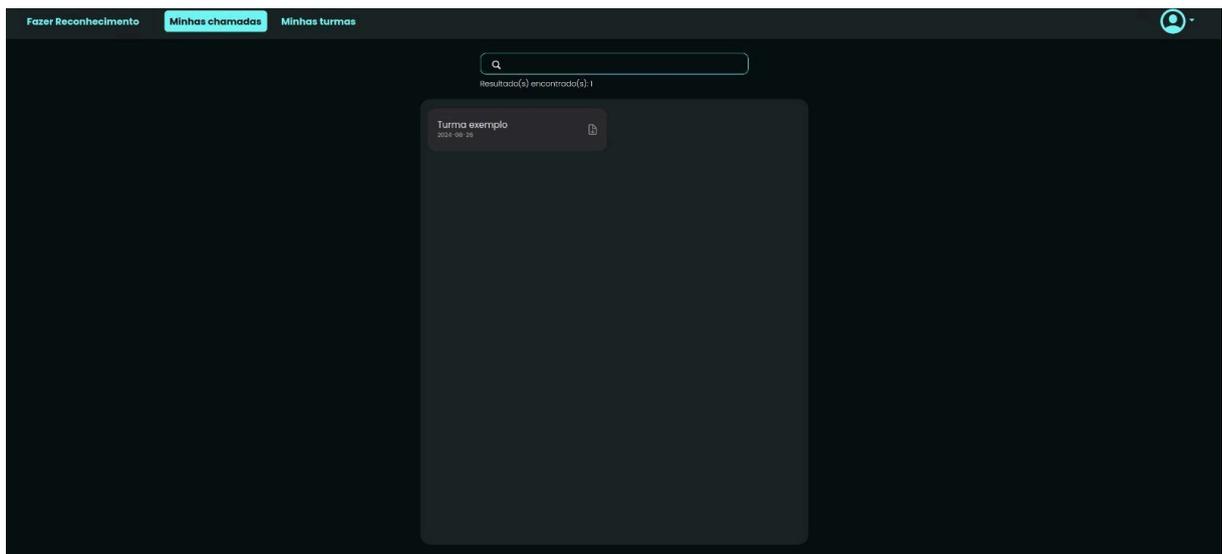


Fonte: autoria própria

3.4.8. Chamadas Realizadas

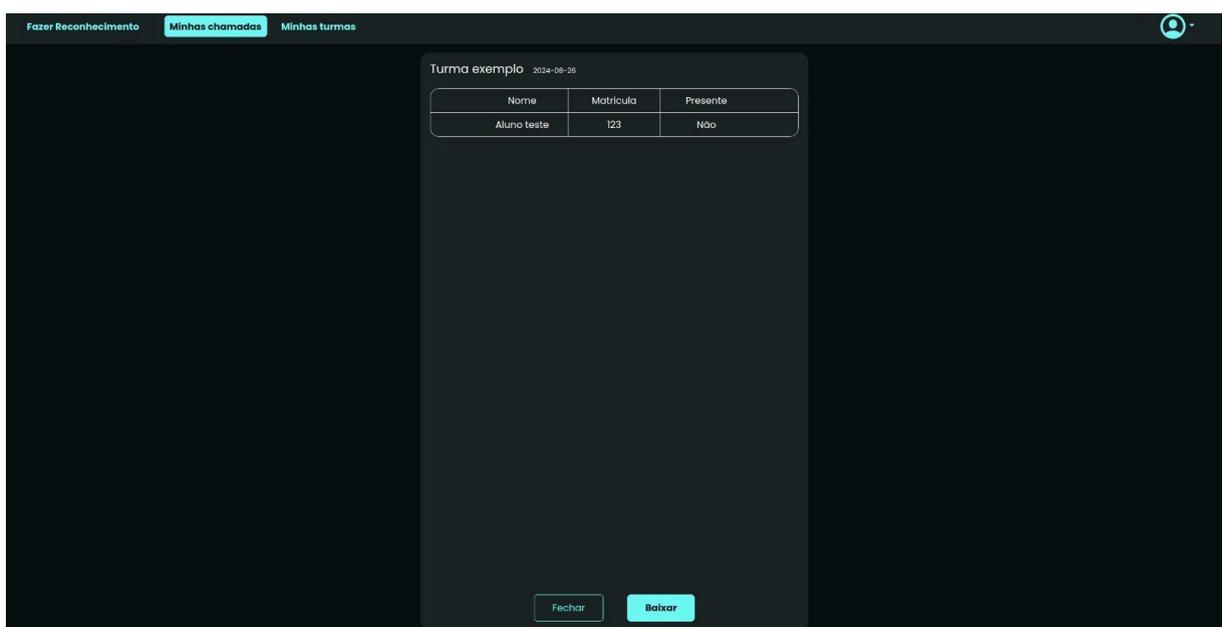
Nestas telas, o usuário pode acessar o histórico de chamadas realizadas, visualizando detalhes como data e turma. Além disso, há a opção de pré-visualizar e baixar um arquivo CSV gerado com os dados da chamada, facilitando o gerenciamento e a análise de presença dos alunos.

Figura 16 – Tela de chamadas realizadas



Fonte: autoria própria

Figura 17 – Pré-visualização do CSV da chamada



Fonte: autoria própria

3.5. USO DE CONTÊINER

Após o desenvolvimento da versão piloto 1.0, foi constatado que a instalação do ambiente de desenvolvimento era complexa e demorada. Para otimizar e simplificar esse processo, optou-se pelo uso de contêineres através do Docker. Essa abordagem permitiu gerenciar e configurar o ambiente de forma automatizada, garantindo maior agilidade na instalação e consistência nas configurações, além de reduzir problemas relacionados à incompatibilidade de dependências.

3.6. CONFIGURAÇÃO DO HARDWARE UTILIZADO PARA OS TESTES

Este tópico descreve o hardware utilizado na realização dos testes do sistema, tanto o servidor que processou os dados quanto o dispositivo que capturou as imagens e vídeos.

Servidor:

- **Processador:** 12th Gen Intel(R) Core(TM) i5-12500H 2.50 GHz;
- **Memória RAM:** 16,0 GB DDR5;
- **Placa de vídeo:** NVIDIA RTX 3050 4GB GDDR6;
- **Armazenamento:** 1TB SSD NVME M.2 (3500 MB/s).

Dispositivo para Captura de Imagens e Vídeos:

- **Modelo:** Samsung Galaxy S20 FE;
- **Câmera traseira:** 12 MP (*wide*), 8 MP (*telephoto*), 12 MP (*ultrawide*);
- **Capacidade de gravação de vídeo:** 4K a 30/60fps, 1080p a 30/60/240fps, 720p a 960fps.

3.7. MÉTRICAS UTILIZADAS NOS TESTES

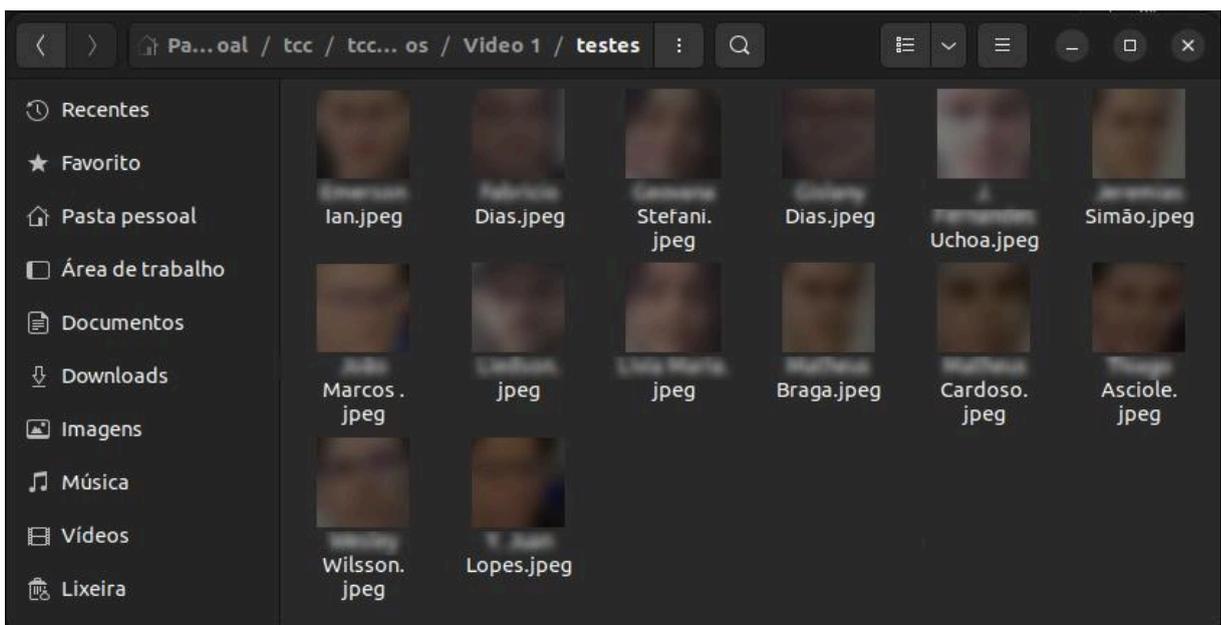
Durante a avaliação do sistema, foram utilizadas diversas métricas para medir o desempenho e a precisão do reconhecimento facial em ambas as versões do sistema. As principais métricas adotadas foram:

- **Tempo de Processamento:** Mede o tempo total necessário para o sistema processar um vídeo ou conjunto de imagens e retornar os resultados. Essa métrica é fundamental para avaliar a eficiência do sistema, especialmente em cenários de uso real, onde tempos de resposta rápidos são críticos. A coleta dessa métrica foi realizada utilizando

o painel "Rede" do navegador, observando-se especificamente a coluna "Tempo" correspondente à requisição de reconhecimento facial.

- **Detecções Corretas:** Refere-se ao número de rostos de alunos identificados corretamente pelo sistema. Essa métrica é essencial para avaliar a precisão do reconhecimento facial, garantindo a confiabilidade do sistema em seu uso prático. A função recognition gerou uma pasta (Figura 18) para cada processamento, contendo as imagens dos rostos dos alunos detectados. O nome de cada arquivo correspondia ao nome do aluno reconhecido. A partir dessas informações, foi realizada manualmente a separação entre detecções corretas e incorretas.
- **Detecções Incorretas:** Contabiliza o número de falsos positivos, ou seja, casos em que o sistema identifica incorretamente um rosto, associando-o a um aluno diferente ou identificando um rosto inexistente. Esta métrica ajuda a identificar falhas no algoritmo de reconhecimento e a necessidade de ajustes.
- **Total de Faces Visíveis:** Refere-se ao número total de rostos que estavam visíveis no vídeo ou nas imagens e que o sistema deveria detectar. Essa métrica é fundamental para calcular a taxa de acerto do sistema, servindo como referência para contextualizar as detecções corretas e incorretas. A contagem das faces visíveis foi realizada manualmente a partir dos vídeos capturados, fornecendo uma base para a avaliação do desempenho do sistema em relação ao total de alunos presentes.

Figura 18 – Pasta gerada para os testes



Fonte: autoria própria

3.8. TESTE DA VERSÃO PILOTO 1.0 DO SISTEMA

Após o desenvolvimento da versão piloto 1.0 do sistema, que utiliza reconhecimento facial baseado em vídeos, foi realizado um teste piloto para verificar seu funcionamento. Esse teste controlado envolveu a utilização de fotos de dois alunos, que foram cadastrados no sistema, e um vídeo capturando exclusivamente esses mesmos alunos. A Figura 19 e a Figura 20 mostram as imagens usadas para o cadastro, enquanto a Figura 21 faz uma representação do vídeo gravado. Esse teste permitiu avaliar a precisão e a eficácia do sistema em um ambiente controlado, fornecendo resultados valiosos sobre o desempenho do reconhecimento facial em situações ideais.

Figura 19 – Foto de registro do aluno 1

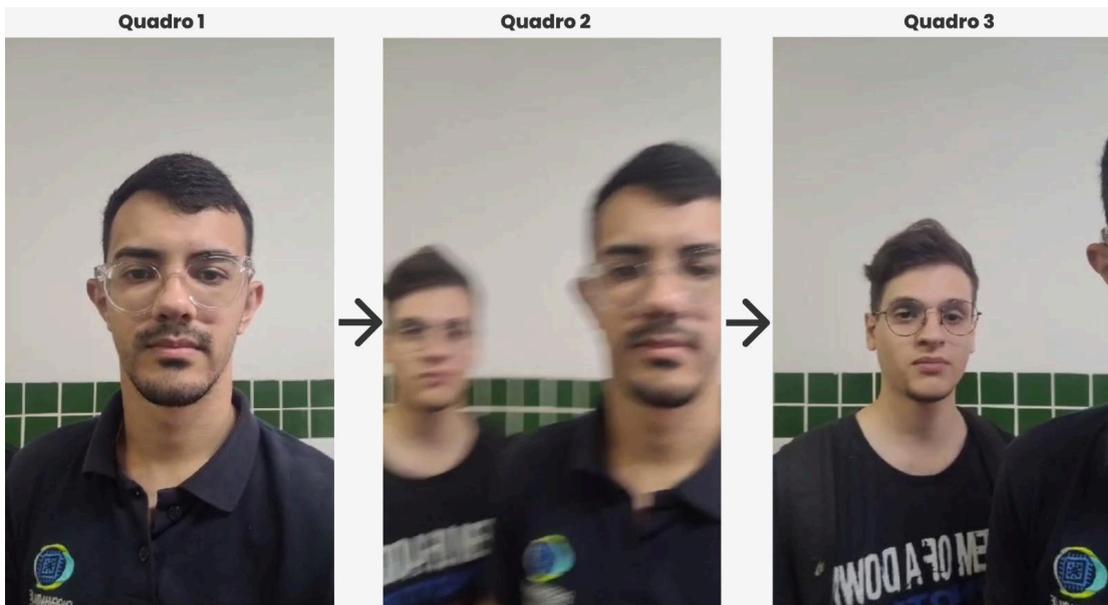


Fonte: Autoria própria

Figura 20 – Foto de registro do aluno 2

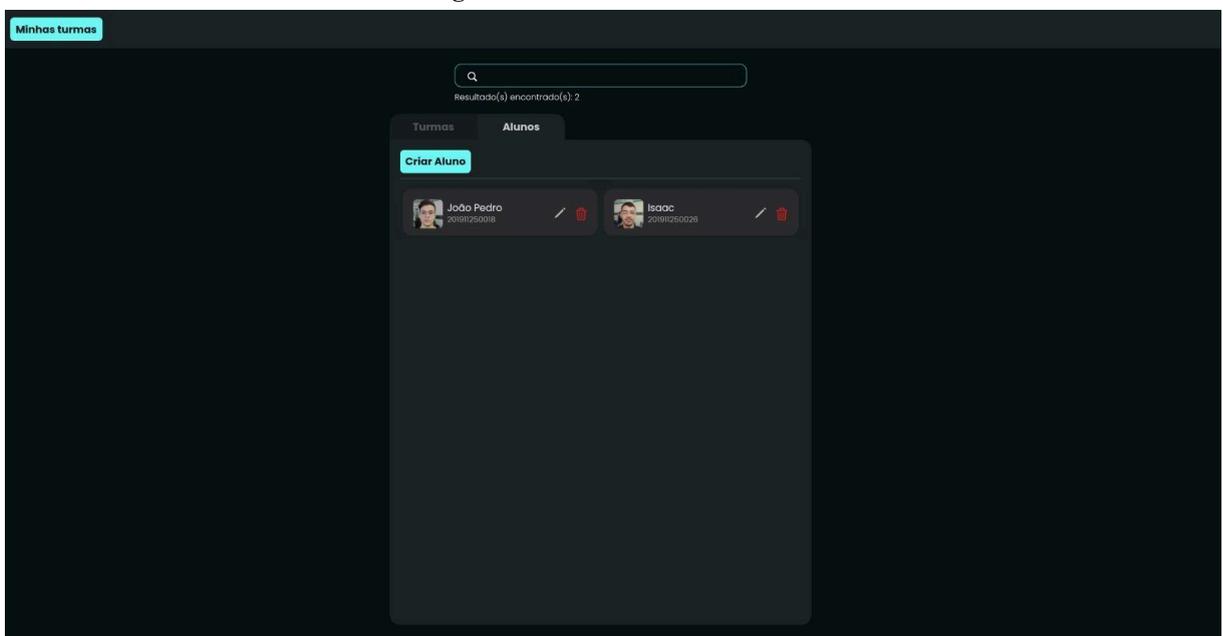


Fonte: Autoria própria

Figura 21 – Captura de tela do vídeo teste

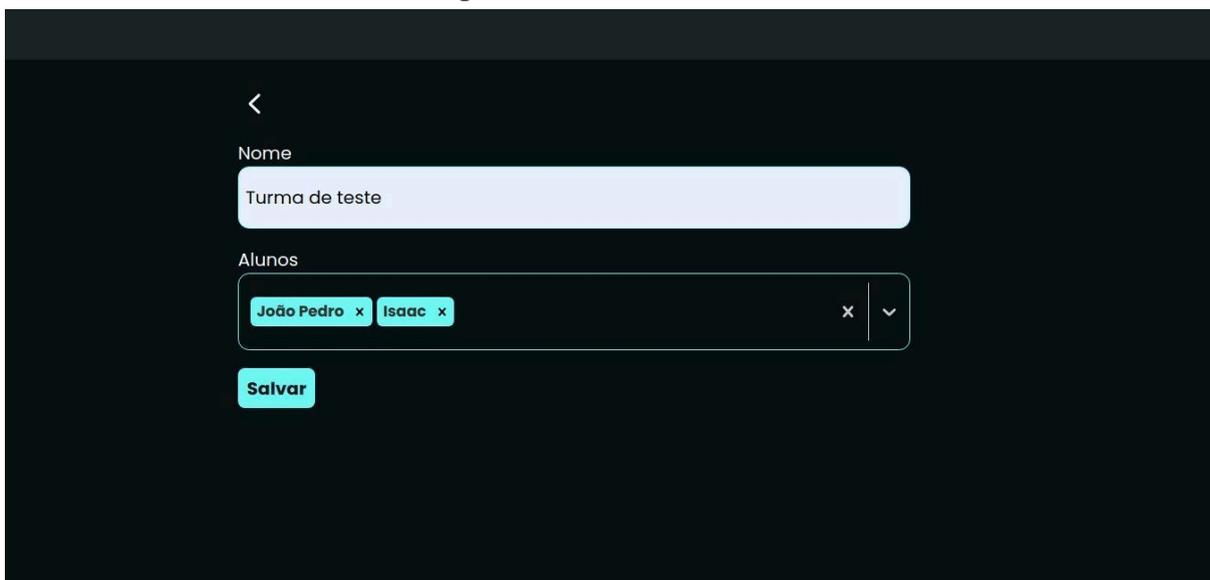
Fonte: autoria própria

Com essas imagens obtidas (Figuras 19 e Figura 20), os alunos foram cadastrados no sistema (Figura 22) e posteriormente adicionados a uma turma (Figura 23).

Figura 22 – Alunos cadastrados

Fonte: autoria própria

Figura 23 – Turma cadastrada



Nome

Turma de teste

Alunos

João Pedro x Isaac x

Salvar

Fonte: autoria própria

Após o registro dos alunos e a criação da turma, o sistema passou a permitir a realização do reconhecimento facial para contabilizar a presença dos alunos. Utilizando o vídeo representado na Figura 21 e a turma criada, intitulada "Turma de teste", foi possível testar o sistema. A Figura 24 mostra o *checklist* com o resultado do reconhecimento, enquanto a Figura 25 apresenta o arquivo CSV gerado com a lista de presença.

Figura 24 – Checklist com resultado do reconhecimento

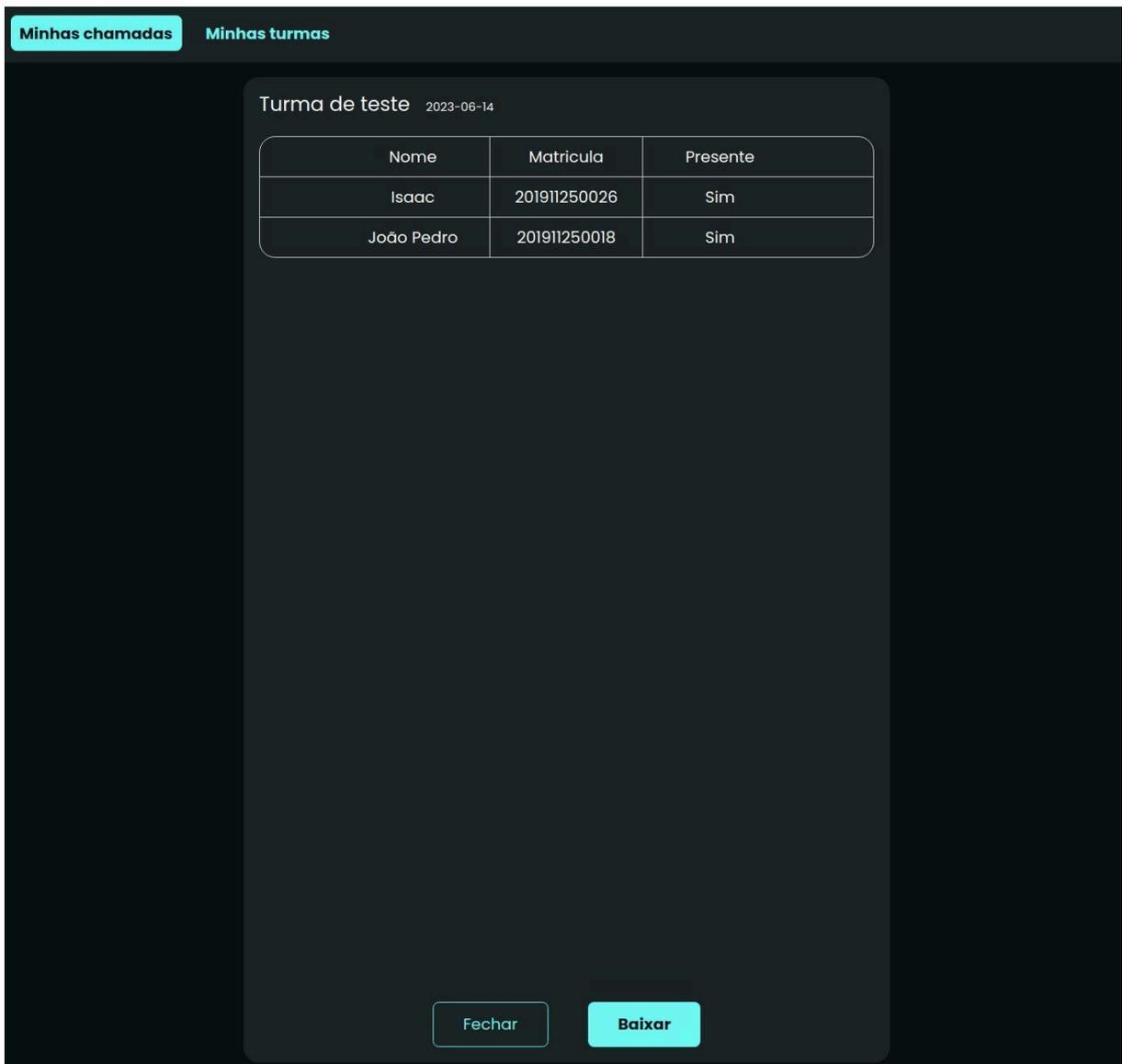
The image shows a dark-themed user interface for a checklist. At the top, there are two categories: 'Alunos presentes' (Students present) indicated by a green dot, and 'Alunos ausentes' (Students absent) indicated by a red dot. Below these, there are two rounded rectangular boxes. The first box, under 'Alunos presentes', contains the name 'João Pedro'. The second box, under 'Alunos ausentes', contains the name 'Isaac'. At the bottom center of the interface is a cyan button labeled 'Salvar CSV'.

Alunos presentes	Alunos ausentes
João Pedro	Isaac

Salvar CSV

Fonte: autoria própria

Figura 25 – CSV de presença de alunos



Minhas chamadas Minhas turmas

Turma de teste 2023-06-14

Nome	Matricula	Presente
Isaac	201911250026	Sim
João Pedro	201911250018	Sim

Fechar Baixar

Fonte: autoria própria

Com os resultados positivos obtidos neste teste inicial, avançamos para a etapa seguinte: teste do sistema em um ambiente real.

3.9. TESTE DA VERSÃO PILOTO 1.0 DO SISTEMA EM AMBIENTE REAL

Para os testes iniciais, o Professor Dr. Katyusco de Farias Santos gentilmente disponibilizou um tempo de aula da turma de "Padrões de Projetos". Esse período foi utilizado para a coleta de dados, cadastro dos alunos no sistema e gravação da turma para fins de reconhecimento facial.

A coleta de dados foi realizada em duas etapas principais. Na primeira etapa, foram capturadas fotografias individuais dos alunos, utilizando um dispositivo móvel, que foram posteriormente cadastradas no sistema para compor o banco de dados inicial destinado ao reconhecimento facial. Esta fase teve como objetivo a criação de uma base de dados essencial para a avaliação do algoritmo de reconhecimento.

Na segunda etapa, foram realizadas gravações em ambiente real de sala de aula, gerando um total de quatro vídeos, cada um capturando uma parte da turma. Dessa forma, a junção dos quatro vídeos abrangeu toda a turma. As gravações, feitas com um dispositivo móvel, foram utilizadas para simular cenários reais e testar a eficácia do sistema em condições variadas de iluminação, ângulos de visão e movimentação dos alunos. Essa abordagem possibilitou uma avaliação mais precisa do desempenho do sistema em situações semelhantes às que seriam encontradas em seu uso cotidiano.

Para simular um dispositivo de baixo custo, as gravações foram realizadas em resolução HD. As filmagens foram feitas a uma distância aproximada de 1 metro dos alunos e com o dispositivo posicionado a uma altura de 1,70 metros em relação ao chão, de modo a manter um padrão uniforme durante todos os testes. Isso garantiu que o sistema fosse testado em condições realistas e consistentes, refletindo seu possível uso em cenários de baixo custo e infraestrutura limitada.

3.9.1 Consentimento e Participação

Para a realização da coleta de dados, foi selecionada uma turma de ensino superior, composta exclusivamente por alunos maiores de idade. Todos os participantes foram devidamente informados sobre o projeto e assinaram um termo de consentimento, autorizando o uso de suas imagens e dados para fins de teste e validação do sistema de reconhecimento facial. Esse procedimento assegurou a conformidade com as normas éticas e legais vigentes, garantindo a transparência e proteção dos direitos dos participantes.

3.9.2 Resultado do teste da versão piloto 1.0 do sistema

O primeiro teste com a abordagem baseada em vídeo revelou limitações significativas. Especificamente, a qualidade variável dos *frames* capturados do vídeo impactou negativamente a precisão do sistema de reconhecimento facial. Como o reconhecimento facial a partir de vídeos exige a análise de múltiplos *frames* sequenciais, a baixa qualidade de alguns desses frames, devido a fatores como iluminação inconsistente, ângulos desfavoráveis e movimento, comprometeu a capacidade do sistema de realizar comparações precisas e

consistentes entre as imagens faciais. Um exemplo de *frame* desfavorável pode ser observado no segundo quadro da figura 21.

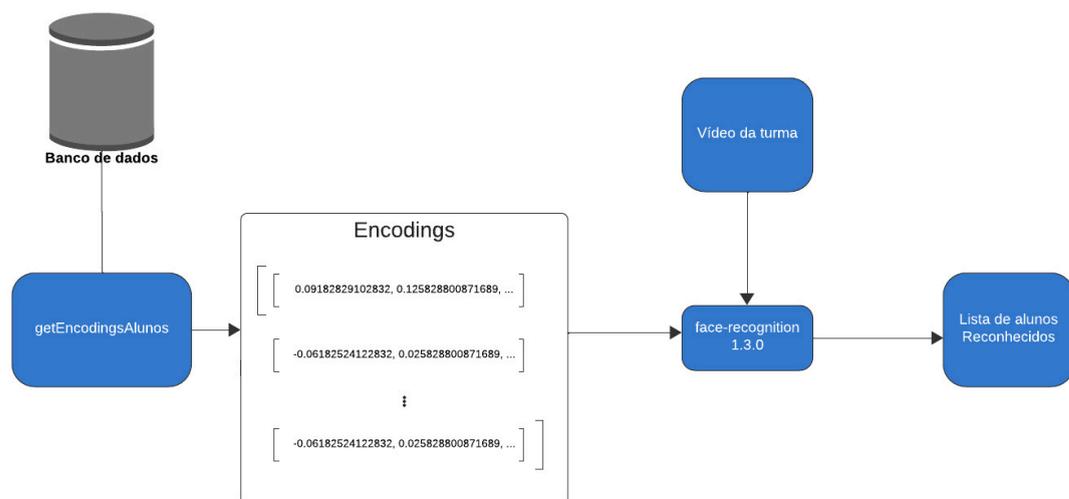
Além disso, a intermitência na qualidade dos frames resultou em tempos de processamento prolongados, uma vez que o sistema precisou realizar múltiplas análises para compensar as inconsistências e tentar garantir a precisão desejada.

Com base nesse teste inicial, foi constatado que a utilização de vídeos não atendeu às nossas expectativas de desempenho e precisão. Em resposta a essas limitações, decidimos explorar a utilização de imagens estáticas como uma alternativa mais estável e confiável para o reconhecimento facial, acreditando que essa abordagem poderia superar as dificuldades encontradas na análise de vídeos.

3.10. DESENVOLVIMENTO DA VERSÃO PILOTO 2.0 DO SISTEMA

Diante das limitações encontradas na versão piloto 1.0, foi necessário realizar modificações significativas na função *recognition* do back-end. A principal mudança foi a substituição do uso de vídeos pelo uso de imagens estáticas para o processo de reconhecimento. Além disso, foi implementada uma otimização no tempo de processamento, ao gerar e armazenar o encoding da fotografia do aluno no momento do cadastro, para uso futuro. Demonstração do novo fluxo do sistema na Figura 26

Figura 26 – Novo Fluxo da Função recognition



Fonte: autoria própria

A função de reconhecimento agora realiza primeiramente uma consulta ao banco de dados para obter os alunos associados à turma selecionada, recuperando diretamente os *encodings* já cadastrados na tabela de alunos.

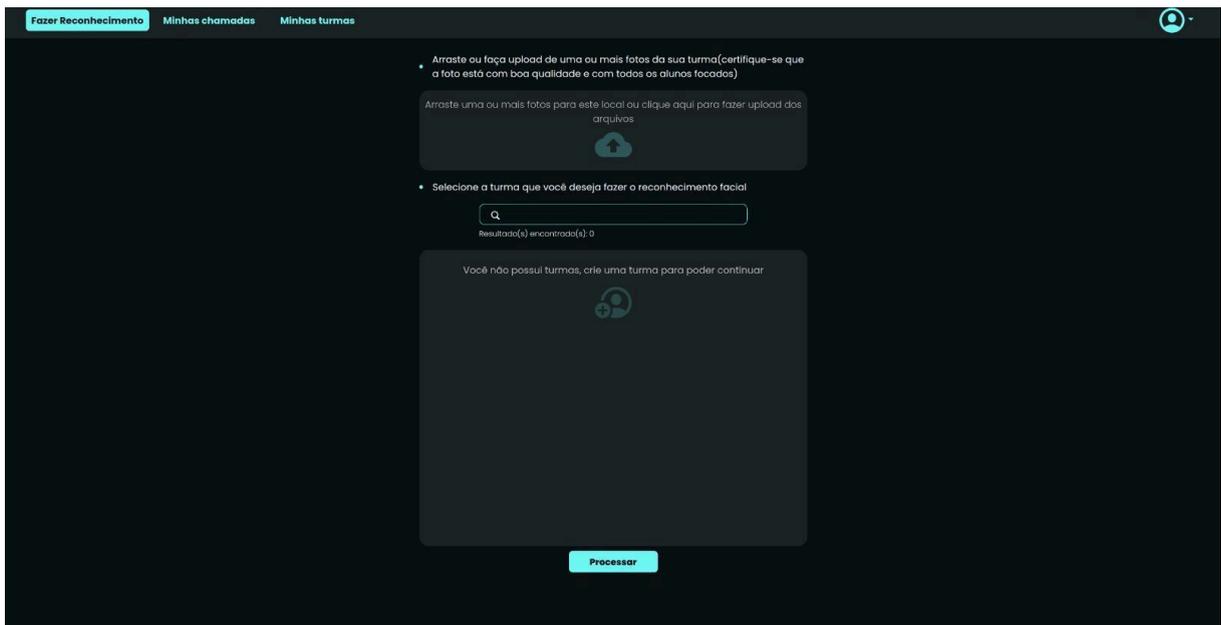
Com o mapa de *encodings* dos alunos pronto, o próximo passo é processar a imagem capturada da turma. Utilize-se a função `face_locations` da biblioteca `face_recognition` para identificar as regiões da imagem que contêm rostos. Em seguida, a função `face_encodings` é aplicada nas regiões detectadas, gerando uma lista de codificações referentes aos rostos presentes na imagem.

Por fim, a comparação entre os *encodings* dos alunos e os *encodings* gerados a partir da imagem é realizada utilizando a função `face_distances`, que calcula a distância entre cada par de *encodings*. Essa comparação permite identificar quais rostos presentes na imagem abrangem os alunos cadastrados, viabilizando o reconhecimento dos presentes na turma.

No front-end, as alterações incluíram a adaptação da interface para permitir o *upload* de imagens em vez de vídeos. Também foi adicionado um compressor de imagens, que otimiza o tamanho do arquivo antes de enviá-lo para o servidor, melhorando a velocidade do *upload*.

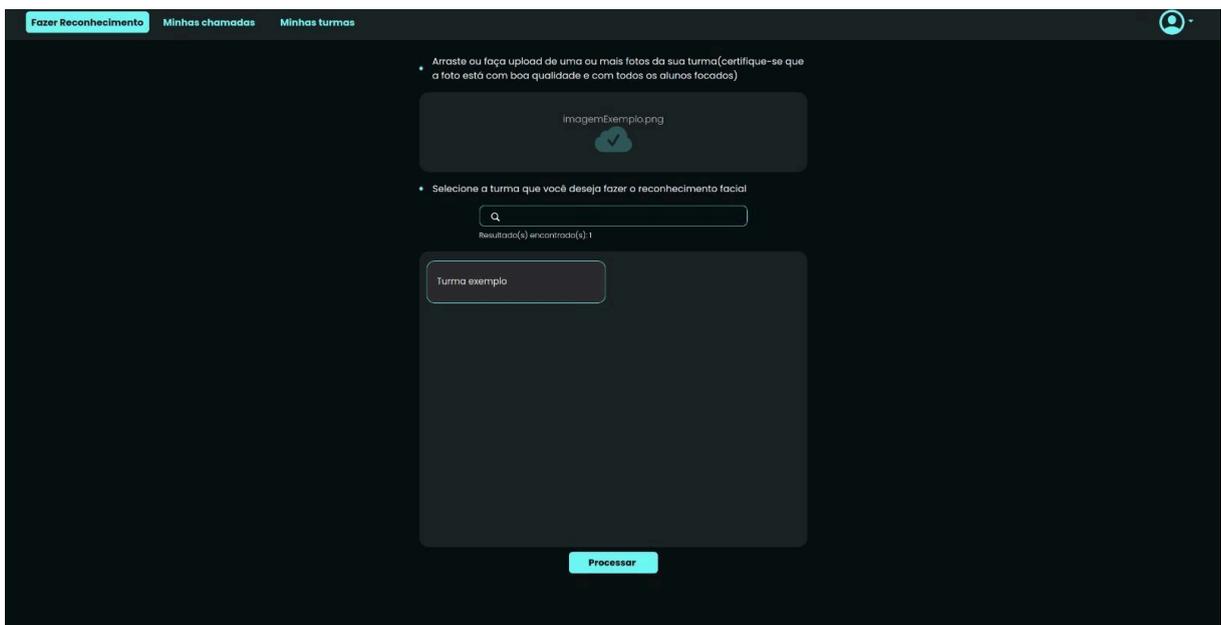
As Figuras 27 e 28 ilustram as mudanças visuais implementadas no frontend. Essas alterações incluem a atualização das instruções de uso, e a modificação do botão "Processar vídeo", que agora está rotulado como "Processar", essas mudanças foram feitas para refletir a nova abordagem do sistema.

Figura 27 – Tela de reconhecimento com imagens



Fonte: autoria própria

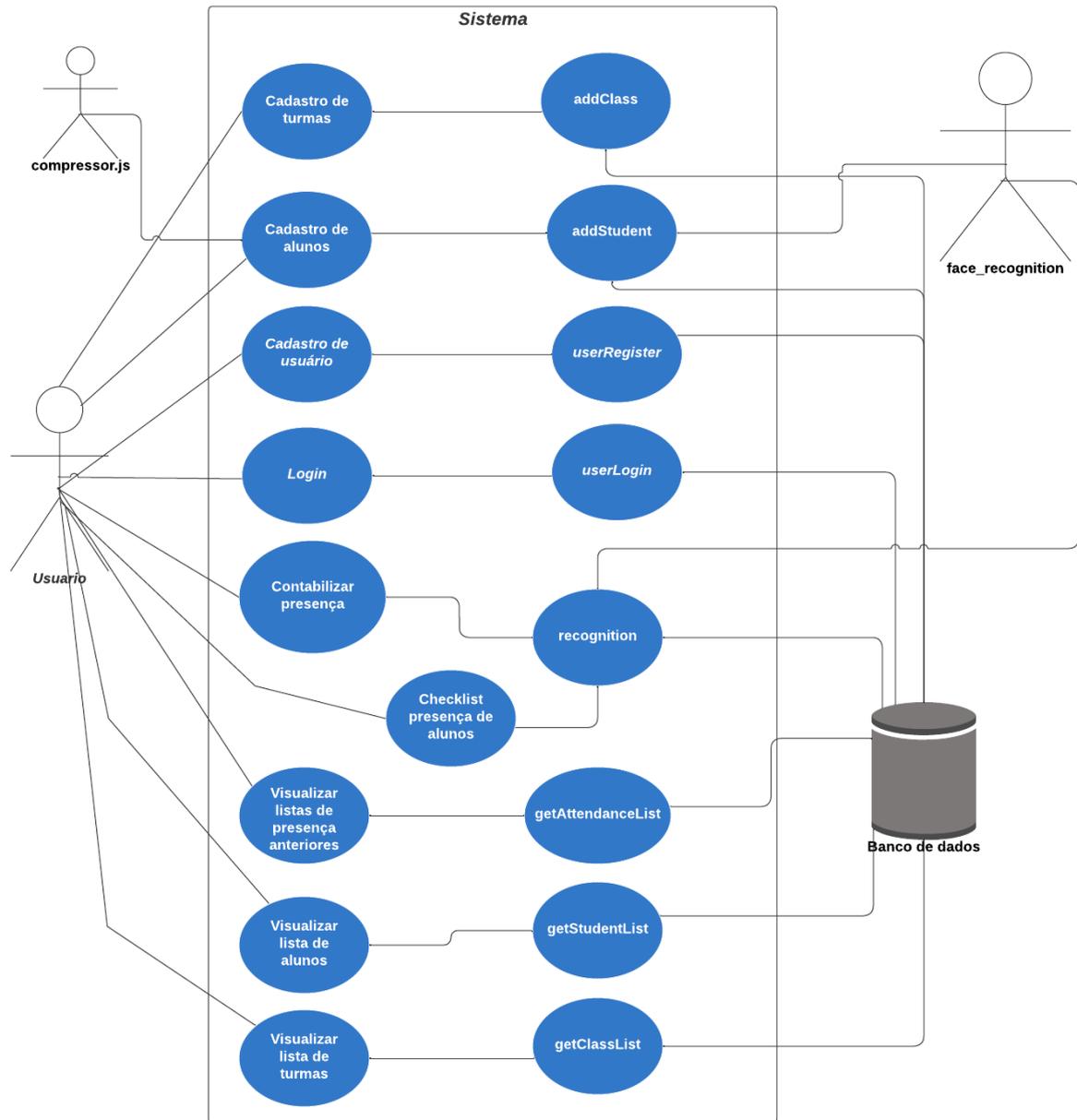
Figura 28 – Tela de reconhecimento com imagens preenchida



Fonte: autoria própria

Com essas alterações, o diagrama de casos de uso também foi atualizado para refletir a nova funcionalidade, incluindo a adição do ator "compressor.js", responsável por realizar a compressão das imagens antes do envio ao servidor. A Figura 29 apresenta o novo diagrama de casos de uso.

Figura 29 – Diagrama de casos de uso da versão piloto 2.0 do sistema



Fonte: autoria própria

3.11. TESTE DA VERSÃO PILOTO 2.0 DO SISTEMA EM AMBIENTE REAL

Após a implementação da nova abordagem, foi realizada uma nova rodada de testes para avaliar a eficácia das modificações. Os mesmos alunos utilizados no teste anterior foram recadastrados na nova versão do sistema, permitindo a geração e o armazenamento dos encodings de seus rostos no banco de dados. Em vez de processar vídeos diretamente, imagens estáticas foram extraídas dos *frames* dos vídeos obtidos na fase anterior de testes, gerando um conjunto de imagens representativas para cada vídeo. Essas imagens foram

selecionadas cuidadosamente, priorizando *frames* limpos e com o máximo possível de faces visíveis. A versão piloto 2.0 do sistema, focada exclusivamente em imagens estáticas, processou essas capturas, proporcionando uma análise mais precisa e uma identificação mais confiável dos alunos, superando as limitações observadas anteriormente com o uso de vídeos.

3.12. LIMITAÇÕES DOS TESTES

Embora os testes do sistema tenham sido realizados em sala de aula, eles não foram conduzidos diretamente pelos professores, que são os usuários finais. Isso se deve principalmente à falta de infraestrutura necessária para realizar o deploy completo da aplicação. O processo de deploy envolve a configuração de servidores e redes que não estavam disponíveis no momento. Por conta dessa limitação, nosso foco foi desenvolver o sistema e garantir sua funcionalidade em ambiente controlado.

Com isso, o deploy foi adiado para trabalhos futuros, quando a infraestrutura adequada estiver disponível. Assim, os testes com os professores, essenciais para validar o sistema no contexto real do dia a dia, ficaram pendentes e serão realizados após a implementação do deploy.

4.0. RESULTADOS E DISCUSSÕES

Na versão piloto 1.0, o sistema foi baseado no processamento de vídeos capturados em sala de aula. Foram obtidos um total de quatro vídeos, e cada vídeo foi testado individualmente. Durante esses testes, foram registradas diversas métricas, incluindo a quantidade de detecções corretas, detecções incorretas, e o tempo de processamento. Além disso, foi calculada manualmente a quantidade de faces visíveis em cada vídeo, proporcionando um contexto mais preciso para a análise dos resultados. Esses dados foram essenciais para avaliar a eficácia do sistema e identificar suas limitações ao lidar com vídeos em ambientes reais.

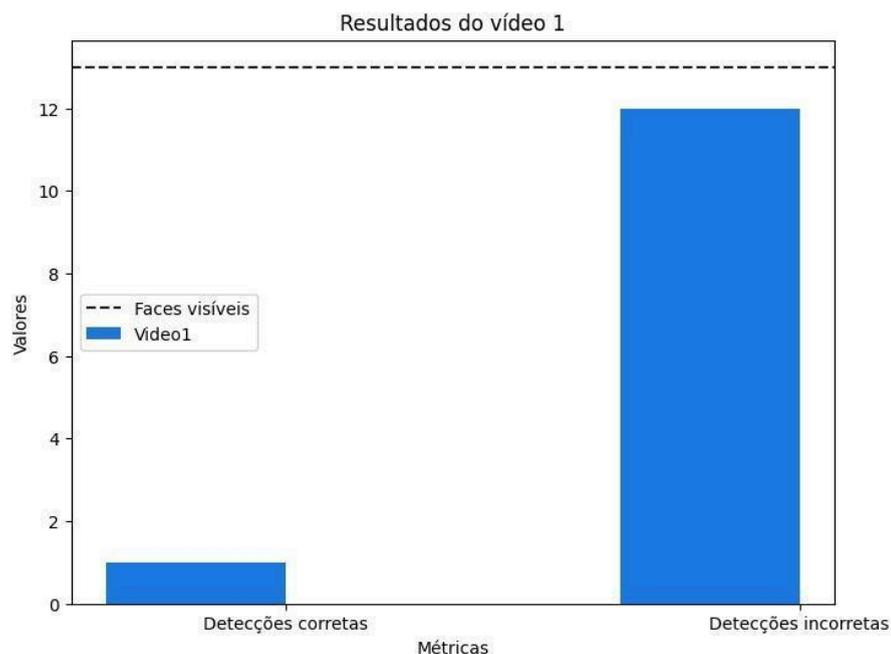
Na versão piloto 2.0, o sistema foi ajustado para processar imagens estáticas extraídas dos vídeos anteriormente capturados. Esse conjunto de imagens foi então submetido à nova versão do sistema, permitindo que fossem registradas métricas semelhantes às da versão piloto 1.0, como detecções corretas, detecções incorretas, e o tempo de processamento.

4.1 RESULTADOS DOS TESTES DA VERSÃO PILOTO 1.0 DO SISTEMA

A seguir, são apresentados os resultados dos testes realizados com a versão piloto 1.0 do sistema, que utilizou vídeos capturados em ambiente real para o reconhecimento facial.

1. Vídeo 1

Figura 30 – Resultados obtidos do vídeo 1

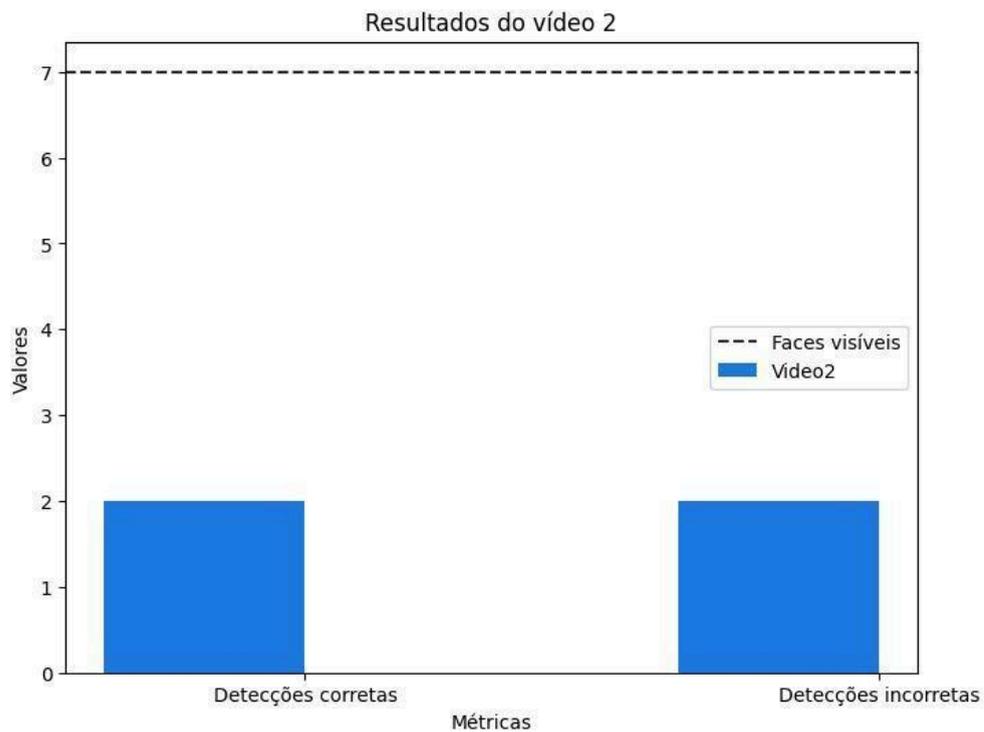


Fonte: autoria própria

- **Número de Alunos com Faces Visíveis:** 13
- **Duração do vídeo:** 10 segundos
- **Dimensão do vídeo:** 480 x 864 px
- **Tamanho:** 3,2 mb
- **Tempo de Processamento:** 49.44 segundos
- **Resultados:** O sistema detectou corretamente apenas um aluno, enquanto houve 12 detecções incorretas. Além disso, o tempo de processamento ultrapassou o limite máximo aceitável de 30 segundos, indicando uma ineficiência significativa ao lidar com vídeos.

2. Vídeo 2:

Figura 31 – Resultados obtidos do vídeo 2



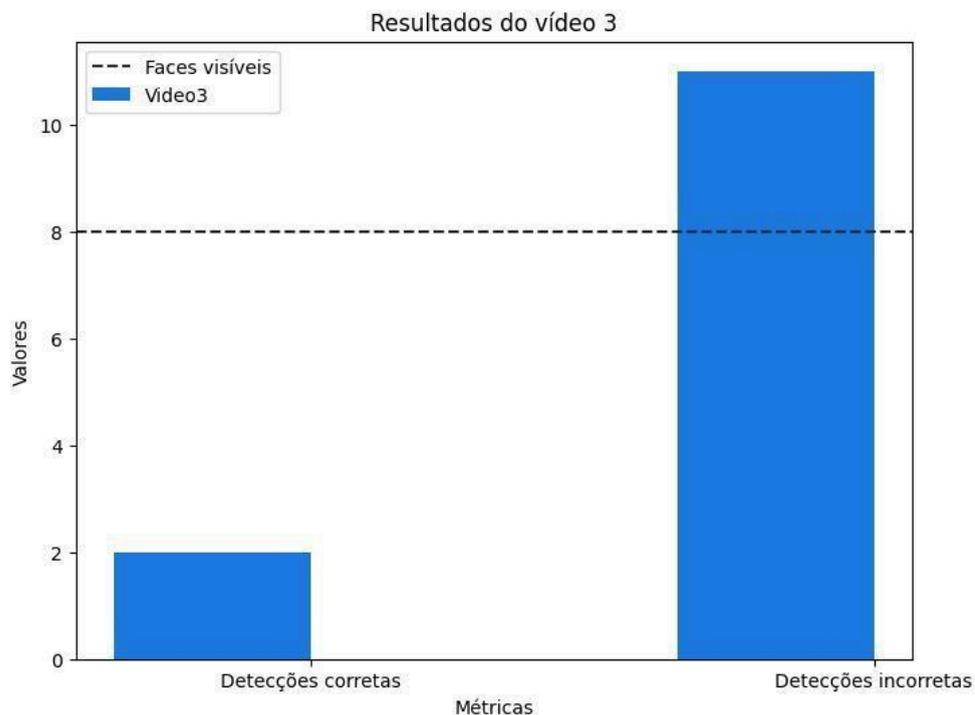
Fonte: autoria própria

- **Número de Alunos com Faces Visíveis:** 7
- **Duração do vídeo:** 11 segundos
- **Dimensão do vídeo:** 480 x 864 px
- **Tamanho:** 3,6 mb
- **Tempo de Processamento:** 2.5 minutos

- **Resultados:** O sistema consegue reconhecer corretamente dois alunos, e dois foram identificados incorretamente. Além disso, o processamento foi mais lento do que o esperado. Estes resultados destacam as limitações da versão piloto 1.0, incluindo o tempo de processamento elevado e a baixa precisão na detecção em ambientes reais.

3. Vídeo 3:

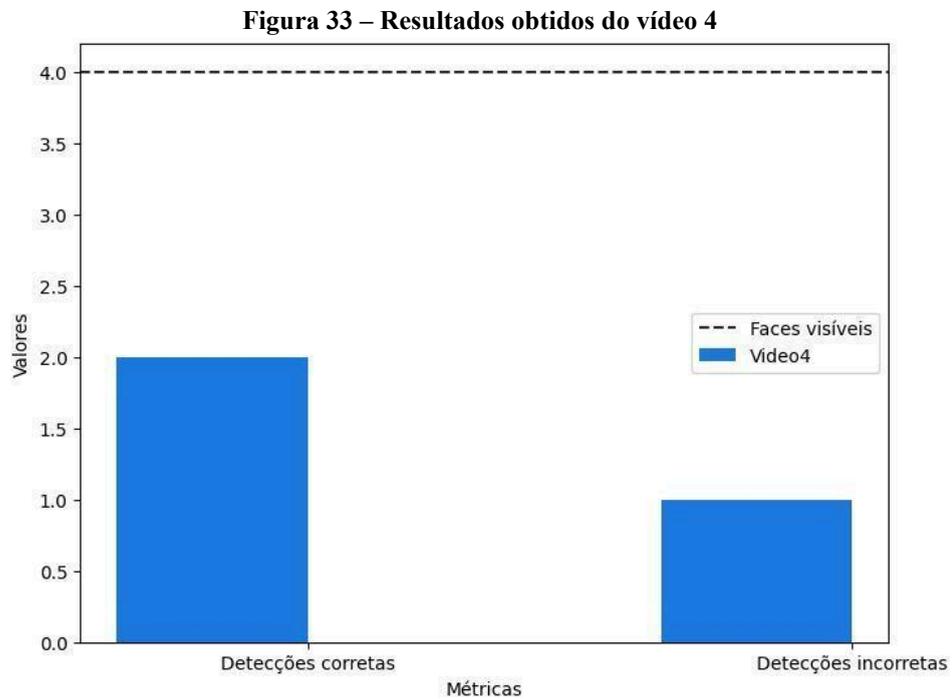
Figura 32 – Resultados obtidos do vídeo 3



Fonte: autoria própria

- **Número de Alunos com Faces Visíveis:** 8
- **Duração do vídeo:** 11 segundos
- **Dimensão do vídeo:** 480 x 864 px
- **Tamanho:** 3,6 mb
- **Tempo de Processamento:** 1.6 minutos
- **Resultados:** O sistema identificou corretamente dois alunos, mas apresentou 11 detecções incorretas, resultando em uma elevada taxa de falsos positivos. Esse desempenho foi impactado pela qualidade inconsistente dos frames capturados no vídeo, o que comprometeu a precisão do reconhecimento facial.

4. Vídeo 4:

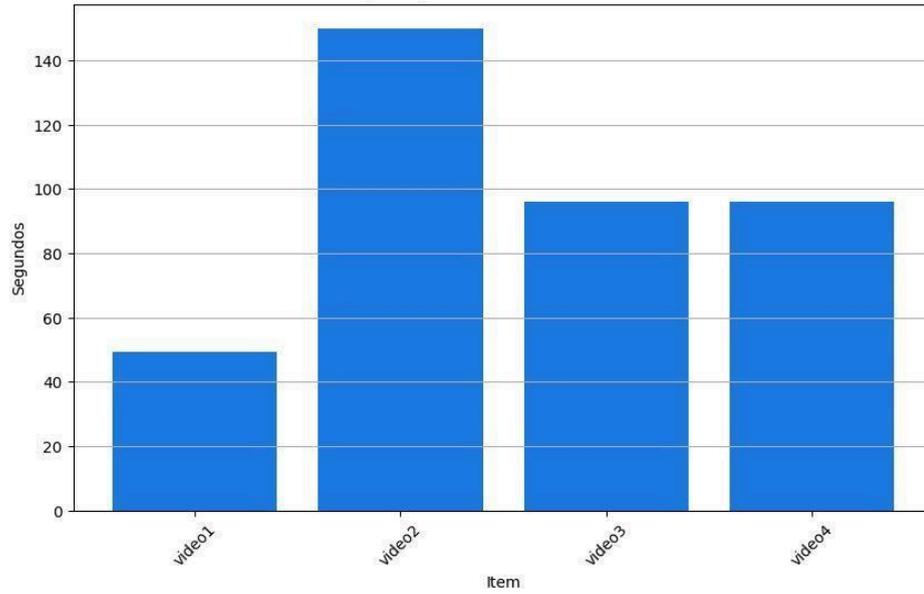


Fonte: autoria própria

- **Número de Alunos com Faces Visíveis:** 4
- **Duração do vídeo:** 8 segundos
- **Dimensão do vídeo:** 480 x 864 px
- **Tamanho:** 3,5 mb
- **Tempo de Processamento:** 1.6 minutos
- **Resultados:** O sistema reconheceu duas pessoas corretamente, mas cometeu um erro de identificação.

O gráfico abaixo (Figura 34) apresenta o tempo de processamento, em segundos, para cada um dos vídeos testados. Os resultados mostram que a versão piloto 1.0 do sistema excedeu significativamente a restrição de 30 segundos.

Figura 34 – Tempo de Processamento por Vídeo na Versão Piloto 1.0 do Sistema



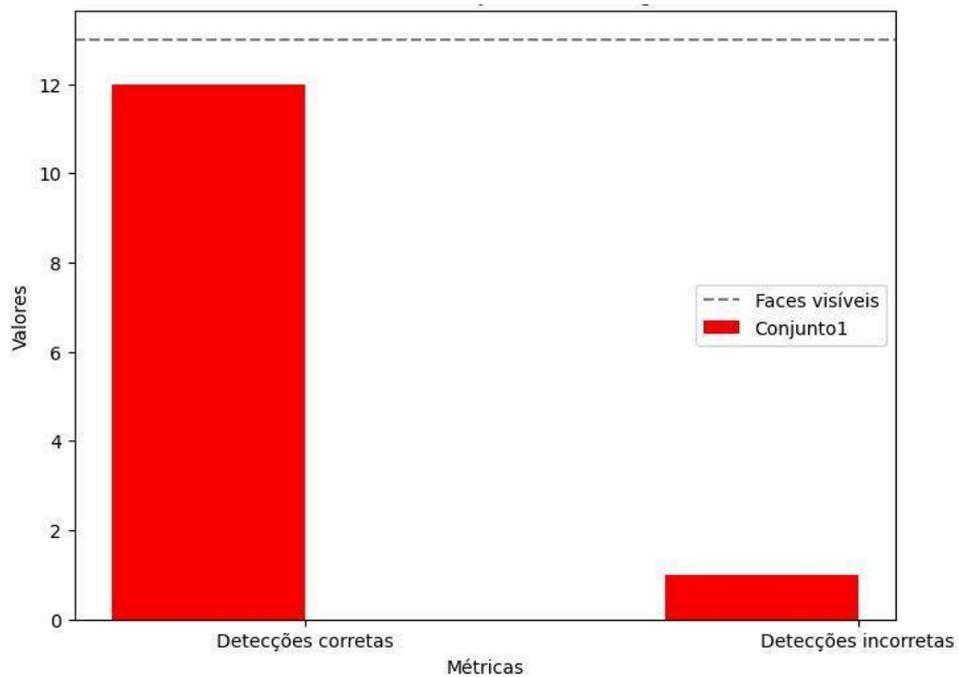
Fonte: autoria própria

4.2 RESULTADOS DOS TESTES DA VERSÃO PILOTO 2.0 DO SISTEMA

A seguir, são apresentados os resultados dos testes realizados com a versão piloto 2.0 do sistema, que utilizou um conjunto de imagens capturadas dos vídeos para o reconhecimento facial.

1. Imagens capturadas do Vídeo 1

Figura 35 – Resultados obtidos das imagens capturadas do Vídeo 1

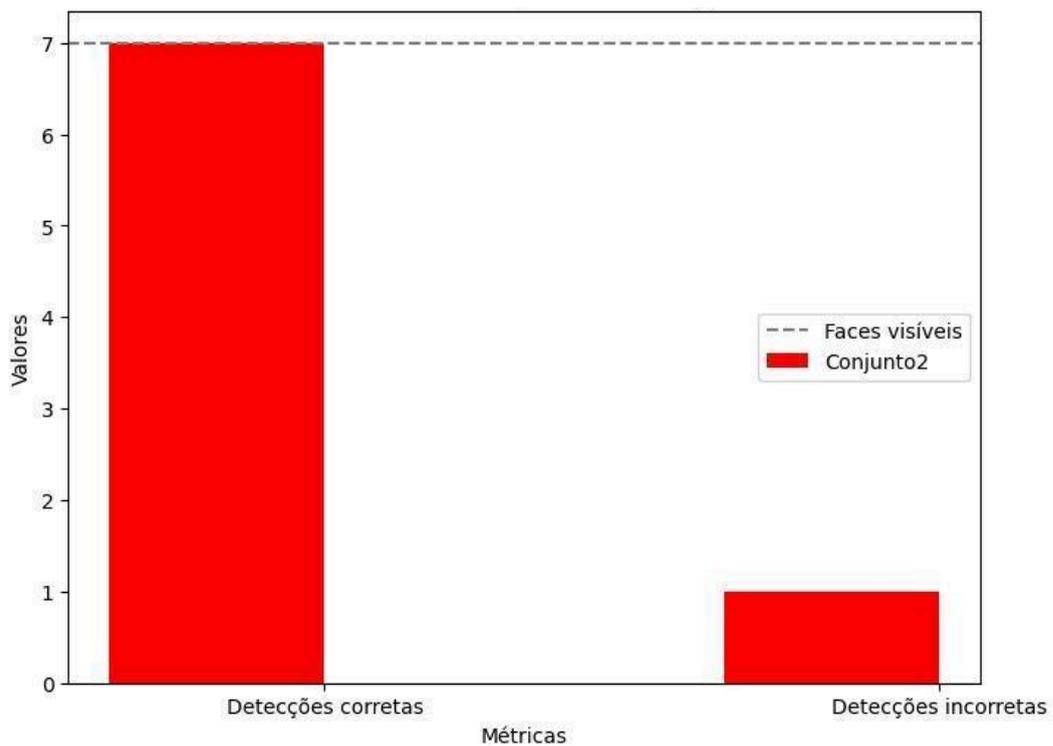


Fonte: autoria própria

- **Quantidade de Capturas Obtidas:** 4
- **Dimensão das capturas:** 480 x 864 px
- **Tamanho:** 115 kb
- **Número de Alunos com Faces Visíveis:** 13
- **Tempo de Processamento:** 2.8 segundos
- **Resultados:** O sistema identificou corretamente 12 alunos, com apenas uma detecção incorreta. Esses resultados demonstram uma precisão significativamente maior em comparação com a abordagem anterior, além de uma redução substancial no tempo de processamento, que agora permanece dentro do limite máximo aceitável de 30 segundos.

2. Imagens capturadas do Vídeo 2

Figura 36 – Resultados obtidos das imagens capturadas do Vídeo 2



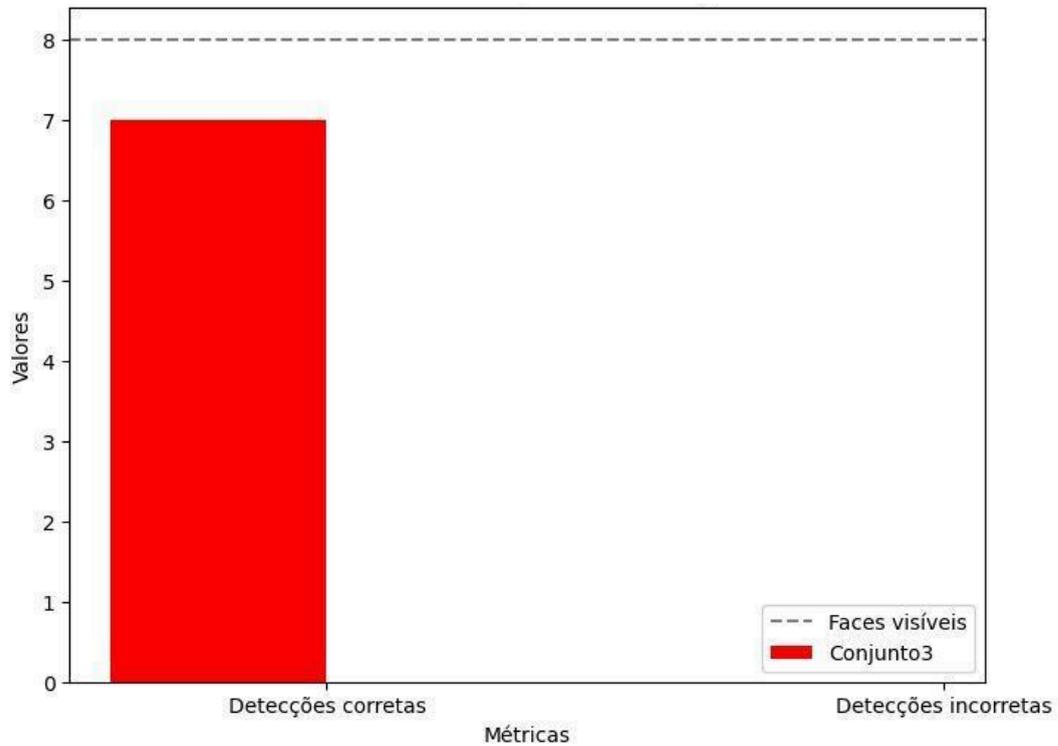
Fonte: autoria própria

- **Quantidade de Capturas Obtidas:** 4
- **Dimensão das capturas:** 480 x 864 px
- **Tamanho:** 115 kb
- **Número de Alunos com Faces Visíveis:** 7
- **Tempo de Processamento:** 2.15 segundos

- **Resultados:** O sistema identificou sete alunos de forma correta, e uma detecção incorreta.

3. Imagens capturadas do Vídeo 3

Figura 37 – Resultados obtidos das imagens capturadas do Vídeo 3

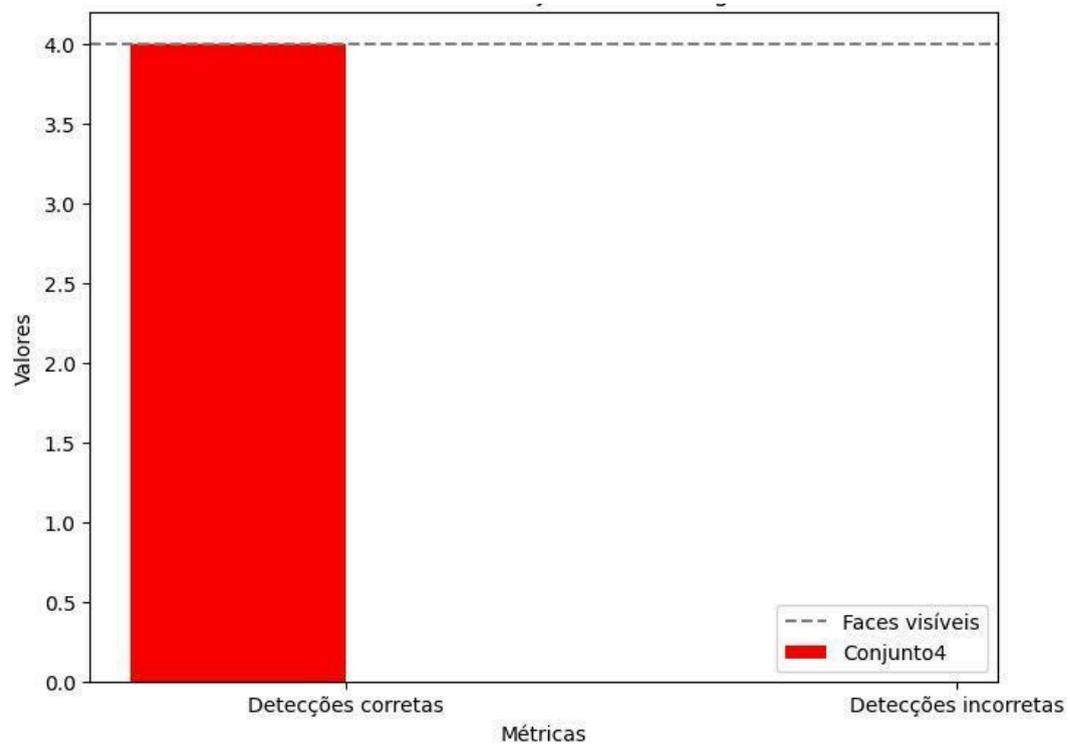


Fonte: autoria própria

- **Quantidade de Capturas Obtidas:** 5
- **Dimensão das capturas:** 480 x 864 px
- **Tamanho:** 115 kb
- **Número de Alunos com Faces Visíveis:** 8
- **Tempo de Processamento:** 2.2 segundos
- **Resultados:** O sistema identificou corretamente sete alunos, sem erros de identificação.

4. Imagens capturadas do Vídeo 4

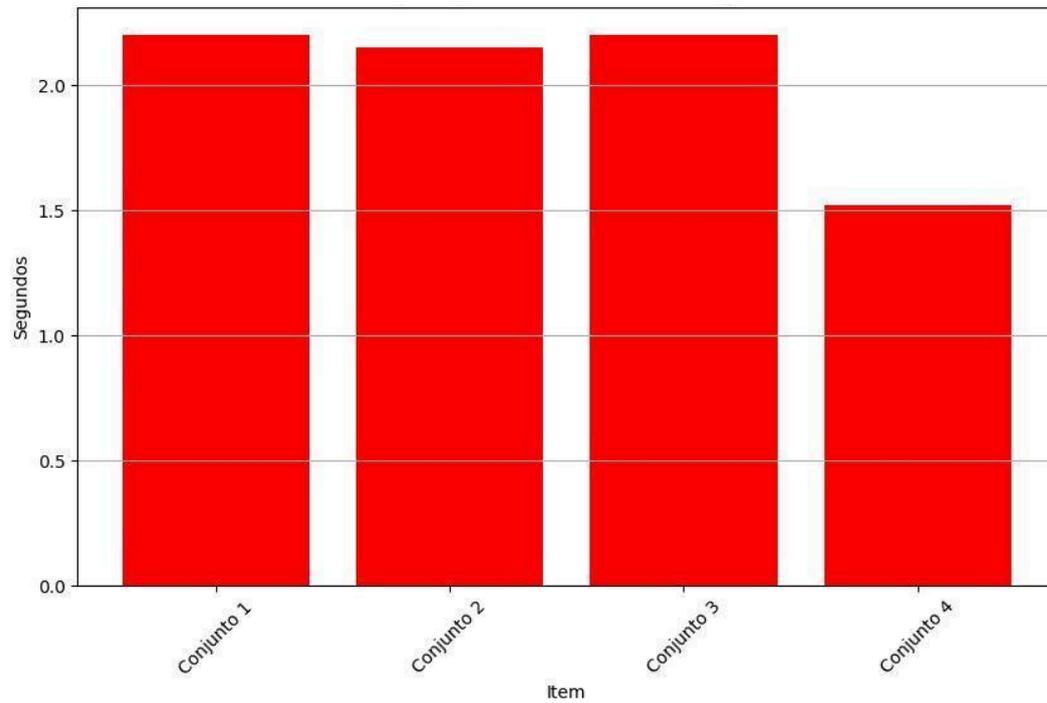
Figura 38 – Resultados obtidos das imagens capturadas do Vídeo 4



Fonte: autoria própria

- **Quantidade de Capturas Obtidas:** 4
- **Dimensão das capturas:** 480 x 864 px
- **Tamanho:** 115 kb
- **Número de Alunos com Faces Visíveis:** 4
- **Tempo de Processamento:** 1.52 segundos
- **Resultados:** : Todos os alunos visíveis nas capturas foram reconhecidos corretamente.

No gráfico abaixo (Figura 39), é apresentado o tempo de processamento, em segundos, para cada um dos conjuntos de imagens testadas. O tempo de processamento atendeu à restrição estabelecida, permanecendo consistentemente abaixo dos 30 segundos.

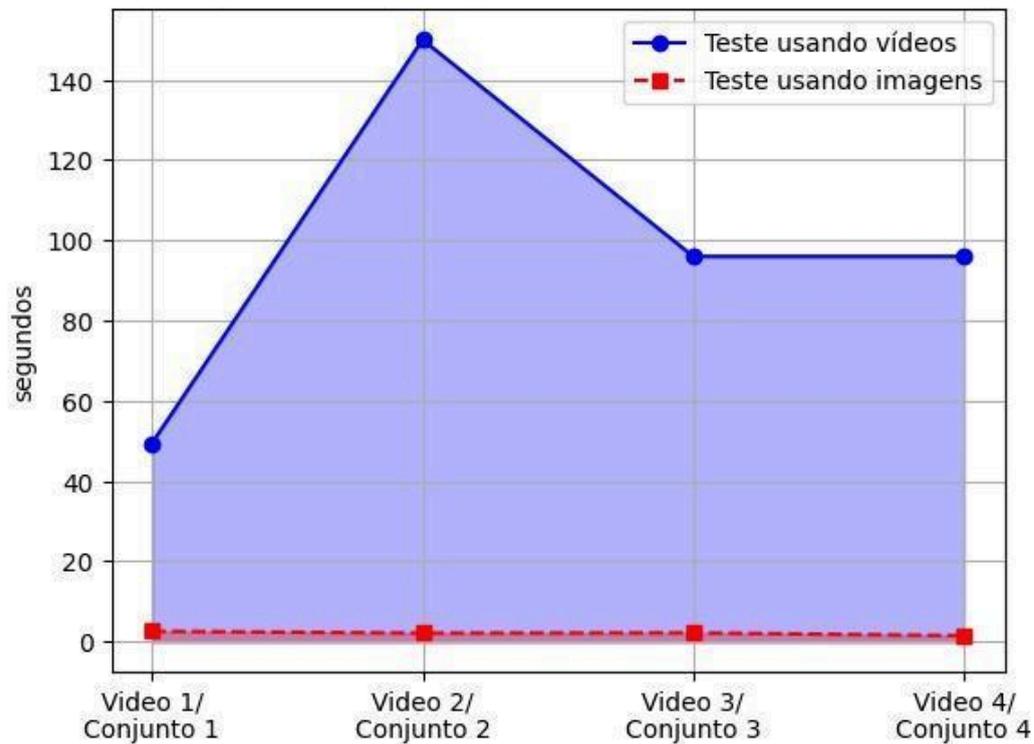
Figura 39 – Tempo de Processamento por Conjunto de imagens na Versão Piloto 2.0 do Sistema

Fonte: autoria própria

4.3 ANÁLISE DOS RESULTADOS

A análise comparativa entre as duas versões do sistema revelou que a nova abordagem, baseada em imagens estáticas, proporcionou uma melhoria significativa tanto na precisão quanto no tempo de processamento do reconhecimento facial. As limitações observadas na abordagem antiga, como o tempo prolongado de processamento e a baixa precisão devido à qualidade variável dos frames, foram mitigadas com a nova abordagem. O gráfico da Figura 40 ilustra a comparação do tempo de processamento entre a versão piloto 1.0 e a versão piloto 2.0 do sistema.

Figura 40 – Comparação dos Tempos de Processamento entre as Versões do Sistema

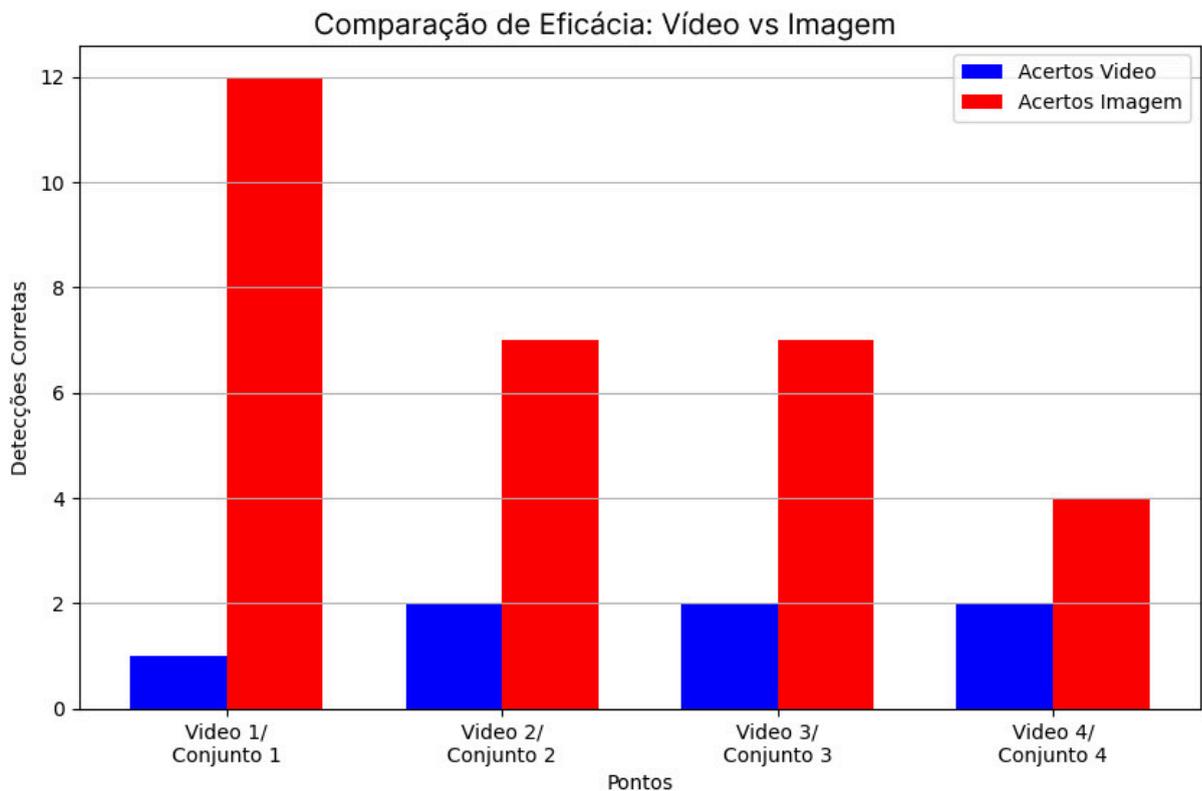


Fonte: autoria própria

O gráfico da Figura 40 destaca uma melhora significativamente aprimorada na versão piloto 2.0 do sistema. Com a utilização de imagens estáticas, o tempo de processamento foi 97,36% mais rápido em comparação com a versão inicial que utilizava vídeos. Essa melhoria demonstra a eficácia da abordagem baseada em imagens para o reconhecimento facial, reduzindo drasticamente o tempo necessário para processar os dados enquanto melhora a precisão do sistema.

Em termos de precisão, a nova abordagem mostrou-se em média 5,25 vezes mais eficiente em relação à versão anterior, reduzindo significativamente a ocorrência de falsos positivos e negativos. O gráfico da Figura 41 ilustra essa melhoria, evidenciando a superioridade da abordagem baseada em imagens estáticas sobre o processamento de vídeos, tanto em termos de precisão quanto de confiabilidade nas detecções.

Figura 41 – Comparação da Eficácia entre as versões do sistema



Fonte: autoria própria

A análise comparativa entre as versões destaca que a segunda abordagem não apenas superou as limitações da primeira, mas também estabeleceu um novo padrão de desempenho para o sistema. A combinação de uma redução significativa no tempo de processamento com um aumento expressivo na precisão torna essa abordagem a mais adequada para utilização em contextos educacionais, garantindo que o sistema possa ser empregado de maneira eficaz em situações que demandem reconhecimento facial confiável.

5.0. CONCLUSÃO E TRABALHOS FUTUROS

A conclusão deste Trabalho de Conclusão de Curso reforça a importância e o impacto da aplicação de tecnologias de reconhecimento facial no ambiente educacional, especificamente no controle de presença dos alunos em sala de aula. O projeto teve como principal objetivo desenvolver uma aplicação capaz de automatizar a chamada de alunos, utilizando um sistema de reconhecimento facial integrado a dispositivos móveis, que se mostrou eficiente ao reduzir o tempo gasto pelo professor com essa atividade.

Ao longo do desenvolvimento, duas abordagens foram testadas. A versão piloto 1.0, que utilizava vídeos para o reconhecimento facial, apresentou limitações significativas, como a baixa precisão e o tempo de processamento elevado, ultrapassando os 30 segundos estipulados como tempo máximo aceitável. Essas limitações evidenciaram a necessidade de ajustes na abordagem, levando à implementação de uma versão piloto 2.0, baseada em imagens estáticas extraídas dos vídeos. Essa nova versão demonstrou maior precisão e eficiência, mantendo o tempo de processamento dentro dos limites estipulados e garantindo uma identificação mais confiável dos alunos presentes.

A aplicação desenvolvida não só atendeu aos requisitos iniciais de otimização do tempo de chamada, mas também garantiu a integridade dos dados e a conformidade com a Lei Geral de Proteção de Dados (LGPD), assegurando que as informações biométricas dos alunos fossem tratadas de maneira segura e ética.

Para trabalhos futuros, algumas melhorias e expansões do sistema podem ser consideradas, tais como:

- **Publicação:** Analisar e escolher a melhor estratégia para disponibilizar o sistema, avaliando as opções de implantação em nuvem ou em servidor local, levando em consideração fatores como custo, escalabilidade, segurança e facilidade de manutenção.
- **Integração com sistemas escolares:** Expandir a aplicação com a integração a sistemas escolares existentes, facilitando o envio automático dos registros de presença para as plataformas acadêmicas utilizadas pelas instituições, como sistemas de gestão de frequência ou portais acadêmicos.

REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965/2014 (Marco Civil da Internet). Diário Oficial da União, Brasília, DF, 15 ago. 2018.

CODIE Newark, About NextAuth.js. Disponível em: <<https://abrir.link/puJsB>>. Acesso em: 30 agosto 2024.

DAVIS E. King, Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009. Disponível em: <<http://dlib.net/>>. Acesso em: 30 agosto 2024.

GEITGEY A., Recognize faces from Python or from the command line. Disponível em: <<https://pypi.org/project/face-recognition/>>. Acesso em: 30 agosto 2024.

GACKENHEIMER, C. Introduction to React. [s.l.] Apress, 2015.

Kruger, B. (2022). O QUE É GIT: CONCEITOS, PRINCIPAIS COMANDOS E QUAIS AS VANTAGENS?. Kenzie. Disponível em: <<https://abrir.link/COAqA>>. Acesso em: 03 março 2024.

LOPES Michele, O que é Figma e como usar?. Disponível em: <<https://abrir.link/BMEYd>>. Acesso em: 30 agosto 2024.

MUI, Material UI - Overview. Disponível em: <<https://mui.com/material-ui/getting-started/>>. Acesso em: 30 agosto 2024.

MURPHY Murphy, The Hidden Dangers of Face Embeddings: Unmasking the Privacy Risks. Disponível em: <<https://ironcorelabs.com/blog/2024/face-embedding-privacy-risks/>>. Acesso em: 30 agosto 2024.

PYTHON foundation, O que é Python?. Disponível em: <<https://abrir.link/XVYDA>>. Acesso em: 30 agosto 2024.

SOUZA Ivan, PostgreSQL: saiba o que é, para que serve e como instalar. Disponível em: <<https://rockcontent.com/br/blog/postgresql/>>. Acesso em: 30 agosto 2024.

SUSNJARA Stephanie, SMALLEY Ian, O que é Docker?. Disponível em: <<https://www.ibm.com/br-pt/topics/docker>>. Acesso em: 30 agosto 2024.

TURK, M.; PENTLAND, A. Eigenfaces for Recognition. Journal of Cognitive Neuroscience, v. 3, n. 1, p. 71–86, jan. 2021.

VILLAIN Mateus; ISABELLE Maria, Figma: o que é a ferramenta, design e como usar. Disponível em: <<https://abrir.link/SyQLJ>>. Acesso em: 30 agosto 2024.

VERCEL, What is Next.js?. Disponível em: <<https://nextjs.org/docs>>. Acesso em: 30 agosto 2024.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

Versão Final do TCC

Assunto:	Versão Final do TCC
Assinado por:	Isaac Barbosa
Tipo do Documento:	Dissertação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Isaac Antônio Silva Barbosa, ALUNO (201911250026) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 03/10/2024 23:05:16.

Este documento foi armazenado no SUAP em 03/10/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1267615
Código de Autenticação: 37625998bf

