

## DESENVOLVIMENTO DE SISTEMA PARA CONTROLE DE FREQUÊNCIA ATRAVÉS DE BIOMETRIA FACIAL

**Gabriel Afonso Pinho de Oliveira<sup>1</sup>, Wallison Franklin Pereira<sup>1</sup>, Rodrigo de Oliveira Plotze<sup>1</sup>**

<sup>1</sup>Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC) Ribeirão Preto, SP – Brasil

oliveira.gap@gmail.com, wallison0076@gmail.com,  
rodrigo.plotze@fatec.sp.gov.br

**Resumo:** *O aumento significativo do uso da tecnologia, impulsionado pela pandemia do COVID-19 e suas restrições, tem estimulado a ampla adoção do processo de reconhecimento facial em diversas áreas. Nesse contexto, o presente artigo tem como objetivo a descrição do desenvolvimento de um sistema para controle de frequência de comparecimento baseado no processo de biometria facial, utilizando o framework Flutter como principal ferramenta de desenvolvimento, com assistência de redes neurais. Com isso, espera-se que a implementação desse sistema traga benefícios significativos aos usuários, como a otimização do tempo e a melhoria geral na precisão e confiabilidade dos dados de frequência dos alunos.*

**Abstract:** *The significant increase in technology usage, driven by the COVID-19 pandemic and its restrictions, has stimulated the widespread adoption of facial recognition technology in various areas. In this context, this article aims to develop a system for attendance frequency control based on facial biometrics, using the Flutter framework as the main development tool, with assistance from neural networks. It is expected that the implementation of this system will bring significant benefits to users, such as time optimization and overall improvement in the accuracy and reliability of student attendance data.*

### 1. Introdução

Com o avanço da tecnologia uma das funcionalidades que está em destaque é o reconhecimento biométrico. Há no momento da escrita deste trabalho seis tipos mais utilizados: biometria digital, biometria facial, reconhecimento por íris, reconhecimento por voz, reconhecimento por retina e por assinatura (CUSTÓDIO, 2007).

Com a expansão das tecnologias focadas em *Internet of Things* (IoT) o reconhecimento facial está sendo empregado em diversos tipos de eletrônicos, desde celulares a câmeras de segurança, para controle de acesso físico (SOUZA, 2019). Esta tecnologia já existe há décadas e possui diversas metodologias eficazes e eficientes já comprovadas.

Este recurso está sendo amplamente utilizado, principalmente, em atividades objetivas e que há necessidade de um processo mais ágil e seguro, ou seja, menos propenso a falhas, por exemplo no acesso a um determinado setor restrito de uma empresa via câmeras de segurança. Por ser facilmente implementado e necessitar de pouco investimento de recursos de mão de obra e hardware, sua empregabilidade e escalabilidade ganham força.

Outro fator que impulsionou a utilização da tecnologia em tarefas rotineiras foi a pandemia do COVID-19, em que houve grandes restrições de atividades nas rotinas humanas pessoais e profissionais, acelerando muito o investimento nesta tecnologia, principalmente por parte de grandes empresas, para driblar os impedimentos impostos por questões de saúde pública e minimizar o contato direto entre funcionários (BECK, 2021).

Embora existam métodos facilmente utilizados, há grande esforço da comunidade desenvolvedora para zelar pela segurança, uma vez que é imprescindível que sejam respeitados os pilares da segurança da informação, pois o uso de dados pode gerar problemas de ordem até mesmo judiciais (COSTA e OLIVEIRA, 2019). Assim como, para aperfeiçoar os métodos já empregados em ambientes não controlados e com grande demanda, por exemplo locais com grande fluxo de pessoas e mudanças na exposição a luz. Com base nisso, o presente estudo tem como objetivo o desenvolvimento de um sistema nativo para *mobile* com finalidade de controle da frequência dos alunos do curso de análise e desenvolvimento de sistemas da Fatec de Ribeirão Preto.

Este artigo contém seis seções, são elas: Introdução, que contém uma breve contextualização sobre o tema aqui abordado. Referencial bibliográfico, que explana sobre informações técnicas que são imprescindíveis para a compreensão do estudo e dão embasamento teórico ao trabalho. Materiais e métodos, em que é descrita a metodologia adotada para desenvolvimento do aplicativo, desde a prototipação até os casos de teste de qualidade. Desenvolvimento, seção que é responsável por conter a elucidação sobre a construção dos códigos funcionais utilizados de maneira mais técnica e aprofundada. Resultados, que mostra os resultados obtidos até o momento da escrita deste estudo, detalhando as funcionalidades de cada tela do aplicativo. Por fim, a seção de considerações finais, que contém detalhes do desenvolvimento e elaboração de todo processo, ilustrando experiências vividas pelos desenvolvedores, dificuldades e impressões.

## **2. Referencial teórico**

A seção de referencial teórico deste artigo apresenta definições sobre os principais conceitos utilizados como base para compreensão do artigo e no desenvolvimento do aplicativo, discorrendo sobre biometria, reconhecimento facial e redes neurais.

### **2.1 Biometria**

Conforme Phillips, McCabe e Chellappa (1998), a definição de biometria é: "Uma biometria é um dado observado de um humano, que permite a identificação daquela pessoa" e para biometria ser eficiente, é necessário que tenha as seguintes propriedades:

- a) Universalidade: Todos os membros de uma população sendo identificada devem possuir a biometria.
- b) Unicidade: Assinatura biométrica deve ser diferente para todos os membros da

população.

- c) Invariância: Assinatura biométrica deve ser invariante sob as condições que será coletada.
- d) Resistência: A biometria deve ser resistente a potenciais contramedidas.

Sistemas que identificam e reconhecem pessoas em tempo real são ferramentas importantes quando o objetivo é segurança e controle de acesso físico ou a dados, podendo ser feito por meio de chaves de acesso, senhas ou até utilização de recursos de biometria facial, digital ou por voz (ALEX, et al., 2012; ADAMOVIĆ et al., 2020).

Para Li e Jain (2011), o recurso de reconhecimento facial tem vantagens em comparação a outras modalidades de biometrias, como impressão digital e íris. Além de ser natural e não intrusivo, a mais importante vantagem é que pode ser capturado a distância.

No cenário nacional, um exemplo é a cidade de Florianópolis – Brasil, que recentemente iniciou a utilização de um sistema biométrico de reconhecimento facial para o procedimento de embarque no Aeroporto Internacional de Florianópolis. O sistema começou por meio de um projeto-piloto com passageiros voluntários da empresa aérea LATAM. O objetivo do governo brasileiro é implementar essa tecnologia em todos os principais aeroportos do país (DIÁRIO CATARINENSE, 2020).

## 2.2 Reconhecimento facial

Tecnicamente, o reconhecimento facial se baseia em procedimentos que consistem em identificar padrões faciais tais como formato da boca, rosto, distância entre olhos, dentre outros (SILVA e CINTRA, 2015). Esses procedimentos são realizados em duas etapas: detecção e reconhecimento, visando melhor performance, confiabilidade e segurança. A detecção de face consiste apenas em identificar a área referente à face em uma imagem. Durante seu processo podem ser avaliados vários atributos, por exemplo: forma do rosto, boca e olhos, aparência da face ou a combinação destes (YANG, 2002). Enquanto o reconhecimento de facial ou biometria facial funciona por meio da extração de características pré-definidas de uma determinada imagem com faces (ORVALHO, 2019) que são convertidas em dados e então comparados aos dados armazenados em um banco de dados de treinamentos anteriores. Caso ocorra uma equivalência, significa que o rosto analisado corresponde a alguma pessoa já inserida nesse banco de dados.

A vantagem de se detectar a face, em um primeiro momento, é que após esta fase, a procura pelas características fica limitada apenas em uma determinada região da imagem (LOPES, 2016). Com base nisso, neste trabalho será utilizada para o desenvolvimento do aplicativo as bibliotecas *TensorFlow Lite* e *MobileFaceNet*, que possuem algoritmos de detecção e reconhecimento facial, além de ser amplamente implementada neste tipo de aplicações, mantendo o aplicativo totalmente nativo para *mobile*.

## 2.3 Redes neurais

Redes neurais são modelos computacionais que se baseiam no funcionamento do sistema nervoso humano para realizar tarefas que precisem de processamento e análise das informações de entrada. São compostas por unidades interconectadas chamadas de neurônios artificiais. Outro modelo de redes neurais, são as redes neurais convolucionais

(RNC). Elas são um tipo específico de rede neural projetada para processar dados com uma estrutura de grade, como por exemplo, imagens pois, utilizam camadas convolucionais para extrair características relevantes e reduzir a dimensionalidade dos dados. Por esse motivo, as RNC têm sido amplamente utilizadas em aplicações como reconhecimento de objetos e classificação de imagens (ICMC, 2023).

### 3. Materiais e Métodos

O desenvolvimento do aplicativo *mobile* deste estudo subdivide-se em três etapas. A primeira denomina-se prototipação seguida da parte de desenvolvimento da aplicação, que por sua vez, também se subdivide em duas etapas, sendo elas, o desenvolvimento *backend*, que é responsável por todo o processo de detecção e reconhecimento facial e o desenvolvimento *frontend*, que é a parte da aplicação visível e acessível pelo usuário final. E a última fase é a de qualidade onde foram feitos todos os testes com a aplicação pós desenvolvimento.

A etapa de prototipação é a etapa inicial do desenvolvimento e implementação, nela ocorre todo processo criativo referente às telas que estarão presentes na aplicação final. É nesta fase que se cria a apresentação das telas e fluxo de navegação do aplicativo. Sendo assim, o objetivo neste momento foi criar as telas visando a melhor experiência de usuário e interfaces intuitivas através do uso da ferramenta *Figma*, que é um editor online de design gráfico focado na criação de interfaces gráficas e experiência de usuário com ênfase colaborativa. A ferramenta permite que uma equipe de designers tenha acesso simultâneo e trabalhe em cima de um mesmo projeto ao mesmo tempo, facilitando a vida de times com profissionais em trabalho remoto em diversas localidades diferentes (Filipe Garret, 2021).

Na segunda etapa, foi realizado todo o processo de criação do aplicativo, inicialmente, transformando todas as ideias e *designs* criados anteriormente em códigos funcionais. Foi utilizado o *framework Flutter*, que é uma ferramenta para o desenvolvimento aplicativos para diferentes plataformas – *Android*, *iOS*, *Web*, entre outros – ao mesmo tempo com um único código redigido na linguagem *Dart*, classificada como uma linguagem multiparadigma, ou seja, pode ser implementada respeitando o paradigma utilizado pelo desenvolvedor. Logo, seus principais benefícios são a versatilidade, menor curva de aprendizado e a agilidade (Matheus Alberto, 2022). Juntamente, foi utilizado o recurso de detecção facial da biblioteca *Google ML Kit* que é responsável por levar a experiência em *machine learning* do Google para aplicativos móveis permitindo realizar, dessa forma, a detecção de rostos, de forma rápida e efetiva (GOOGLE DEVELOPERS, 2023).

Para realizar o procedimento de reconhecimento facial, foi utilizada a biblioteca *TensorFlow* em sua versão *Lite* que é uma plataforma para construção e treinamento de redes neurais, que permitem detectar e decifrar padrões e correlações, análogas ao aprendizado e raciocínio utilizados pelos seres humanos (WIKIPEDIA, 2015). O *TensorFlow* foi construído com o propósito de conduzir o aprendizado de máquina e a pesquisa de redes neurais profundas. Sua versão *Lite*, foi projetada para ser leve, com inicialização rápida, sendo compatível com uma variedade de plataformas, incluindo *Android* e *iOS*, além de sua estrutura permitir executar a inferência de modelos *TensorFlow* em dispositivos de móveis. Juntamente com o *MobileFaceNets*, que são conjuntos de modelos de redes neurais convolucionais (RNC), que usam uma média de 1 milhão de parâmetros e são especificamente adaptados para verificação facial em tempo

real de alta precisão em dispositivos móveis e embarcados (Sheng Chen, 2018).

A última etapa do desenvolvimento e implementação do aplicativo deste estudo foi a de qualidade. Ela é considerada essencial para garantir que o aplicativo atenda às demandas necessárias de desempenho, usabilidade, segurança e funcionalidade. Para tanto, foram realizadas uma sequência de testes documentados, detalhando variáveis de entrada com instruções e as saídas possíveis e as esperadas para o funcionamento correto da aplicação. Os casos de testes utilizados foram:

- a) **Validação de cadastro do usuário:** São verificados os valores inseridos pelo usuário como forma de registro no momento do primeiro acesso ao aplicativo;
- b) **Validação de *log in* de usuário cadastrado:** realiza uma verificação feita pelo sistema, utilizando os dados que foram inseridos pelo usuário, com o intuito de verificar se o usuário está ou não cadastrado na base de dados. Caso esteja, o usuário da continuidade no processo de *login* no sistema, caso contrário, é emitida uma mensagem informando o erro ocorrido;
- c) **Validação de cadastro de aluno:** Verifica-se todas as informações referentes ao aluno preenchidas pelo usuário foram feitas corretamente;
- d) **Validação de cadastro de rosto:** consiste na verificação das características extraídas do rosto do aluno no momento do seu cadastro. Com dados devidamente extraídos, o sistema realiza uma verificação no sistema, procedendo com o cadastro caso estejam de acordo com o esperado, caso contrário, uma mensagem de erro será emitida informando o motivo da falha;
- e) **Validação de cadastro de disciplina:** Responsável pela verificação dos valores inseridos pelo usuário para registrar uma disciplina.

Para o armazenamento dos dados necessários nos processos supracitados, foi utilizado o *Cloud Firestore* que é um banco de dados flexível e escalonável para desenvolvimento focado em dispositivos móveis, *Web* e servidores pelo *Firebase* e do *Google Cloud*, além de manter os dados em sincronia em aplicativos cliente usando *listeners* em tempo real e oferecer suporte off-line para dispositivos móveis e *Web*, ideal para aplicativos responsivos funcionarem independentemente da latência da rede ou da conectividade com a Internet (FIREBASE, 2023). Enquanto para armazenamento local, ou seja, no próprio dispositivo *mobile*, o serviço de banco de dados utilizado foi o *SQFLite*, que é um *plug-in* para *Flutter*, com suporte para transações e auxiliar para realizar consultas, inserção, atualização e remoção de dados. É executado em segundo plano em aparelhos *iOS* e *Android* (PUBDEV, 2019).

#### 4. Desenvolvimento

Esta seção descreve as técnicas usadas durante o desenvolvimento, desde a detecção e reconhecimento facial ao armazenamento destes dados, incluindo o uso de modelos para o reconhecimento posteriormente.

Para realizar a detecção facial, o primeiro passo, após acessar a câmera do dispositivo, é determinar se há uma ou mais faces visíveis. Com a API de detecção facial do Kit de ML, é possível detectar rostos em uma imagem, identificar os principais traços

faciais e o contorno dos rostos detectados (GOOGLE DEVELOPERS, 2023).

Na Figura 1 está exemplificado como é feita a instanciação do método responsável pela detecção de faces, em que é feita uma atribuição do objeto *FaceDetector* (linha 23), utilizando parâmetros que determinam o nível de performance (linha 25), se os contornos faciais devem ser detectados (linha 26) e se informações adicionais sobre o rosto serão classificadas (linha 27).

```

23  _faceDetector = FaceDetector(
24      options: FaceDetectorOptions(
25          performanceMode: FaceDetectorMode.fast,
26          enableContours: true,
27          enableClassification: true)); // FaceDetectorOptions // FaceDetector
28  }

```

**Figura 1 - Trecho de código**  
**Fonte: (Autoria própria, 2023)**

Ao abrir a câmera, inicia-se uma transmissão de imagens em tempo real, em que o método apresentado na Figura 1 faz uma verificação constante baseado na imagem atual. Na Figura 2 está demonstrado o uso do método responsável por verificar se há ou não uma face visível (linha 66).

```

61  InputImage _firebaseVisionImage = InputImage.fromBytes(
62      bytes: bytes,
63      inputData: inputData,
64  );
65
66  _faces = await _faceDetector.processImage(_firebaseVisionImage);

```

**Figura 2 - Trecho de código**  
**Fonte: (Autoria própria, 2023)**

Após capturar uma imagem, o método demonstrado na Figura 3 é responsável por utilizar uma função que realiza um pré-processamento na imagem que descarta toda a área exterior a face detectada (linha 52); a imagem resultante é transformada em uma matriz de números (linhas 54, 55, 57, 58), para que possa ser armazenada para futuras consultas.

```

49 void setCurrentPrediction(CameraImage cameraImage, Face? face) {
50     if (interpreter == null) throw Exception('O interpretador é nulo');
51     if (face == null) throw Exception('A face é nula');
52     List input = _preProcess(cameraImage, face);
53
54     input = input.reshape([1, 112, 112, 3]);
55     List output = List.generate(1, (index) => List.filled(192, 0));
56
57     interpreter?.run(input, output);
58     output = output.reshape([192]);
59
60     _predictedData = List.from(output);
61 }

```

**Figura 3 - Trecho de código**  
**Fonte: (Autoria própria, 2023)**

Pela Figura 4 pode-se observar o método responsável por realizar a função de controle de frequência, utilizando os alunos que foram previamente cadastrados. Ao abrir a câmera, o sistema faz todo o processamento listado nas imagens anteriores, utilizando como parâmetro as informações da face que está visível no momento e faz uma comparação entre todas as faces que já foram armazenadas no banco de dados (linhas 110, 115 a 121).

```

107 Future<Student?> _searchResult(List predictedData) async {
108     DatabaseHelper db = DatabaseHelper.instance;
109
110     List<Student> students = await db.queryAllStudents();
111     double minDist = 999;
112     double currDist = 0.0;
113     Student? predictedResult;
114
115     for (Student student in students) {
116         currDist = _euclideanDistance(student.modelData, predictedData);
117         if (currDist <= threshold && currDist < minDist) {
118             minDist = currDist;
119             predictedResult = student;
120         }
121     }
122     return predictedResult;
123 }

```

**Figura 4 - Trecho de código**  
**Fonte: (Autoria própria, 2023)**

## 5. Resultados

Nesta seção, estão apresentados os resultados obtidos durante o desenvolvimento da aplicação. Nos critérios de avaliação dos resultados temos como parâmetros os seguintes padrões observados, a precisão do reconhecimento facial, o desempenho do aplicativo e a usabilidade do sistema. Nas figuras a seguir, temos o funcionamento geral do aplicativo e suas principais funcionalidades.

A usabilidade do sistema foi avaliada por meio de testes de interação com os alunos. Os participantes dos testes destacaram a facilidade de uso do aplicativo, sua interface intuitiva e a simplicidade no cadastro de alunos.

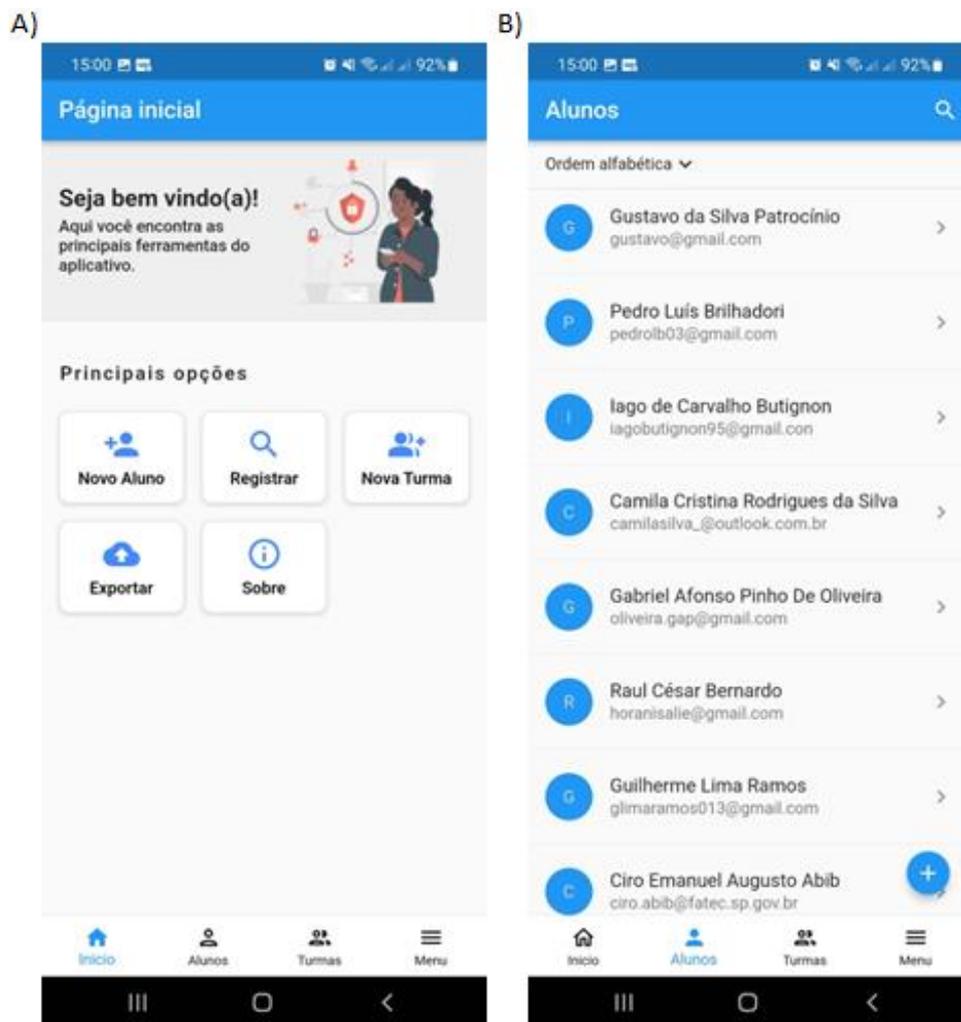


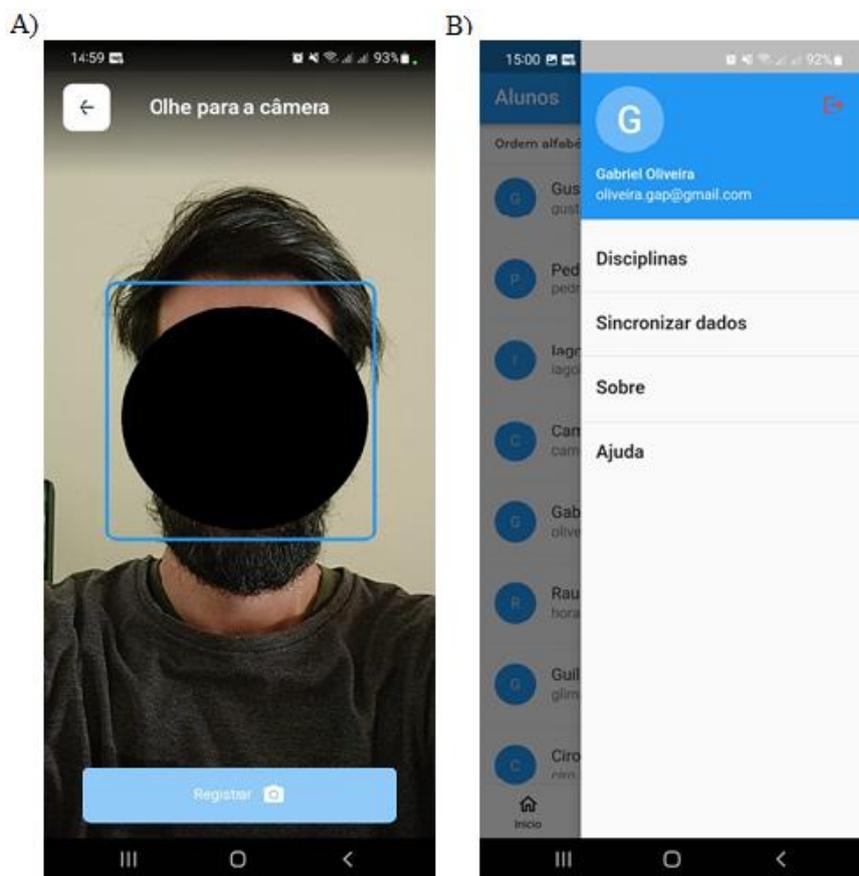
Figura 5 - A) Tela de página inicial B) Tela de alunos cadastrados

Fonte: (Autoria própria, 2023)

Na Figura 5A está apresentada a tela inicial do aplicativo, que possui botões para as principais funcionalidades presentes no aplicativo. Assim, se o usuário clicar no botão ‘Novo Aluno’ será encaminhada a tela de cadastro de alunos. Ao clicar em ‘Registrar’

avançará para tela de reconhecimento facial em que é registrada a presença na aula. No botão ‘Nova turma’, adicionará uma nova turma de alunos.

Figura 5.B) – Tela que exibe os alunos que já estiverem cadastrados no aplicativo, sendo capaz de realizar o procedimento de chamada utilizando a biometria facial do aluno.



**Figura 6 - A) Tela de reconhecimento facial B) Tela de menu lateral  
Fonte: (Autoria própria, 2023)**

Figura 6.A) – Tela de cadastramento da biometria facial através do aplicativo. Ao abrir a câmera, o rosto detectado é delimitado com um quadrado azul, apenas esta região será utilizada pelo algoritmo durante o processo de reconhecimento facial para registrar o comparecimento do aluno.

Figura 6.B) – Tela de menu lateral. Apresenta funcionalidades extras do aplicativo que não são utilizadas constantemente, são funções mais técnicas que devem ser utilizadas pelo administrador do aplicativo. Por exemplo, as disciplinas disponíveis para o registro de frequência dos alunos, informações sobre o aplicativo e opção de ajuda ao usuário.

A precisão do reconhecimento facial foi avaliada utilizando como base de dados, alunos matriculados no quinto semestre do curso de análise e desenvolvimento de sistemas. Os algoritmos de detecção e reconhecimento facial implementados apresentou uma taxa de acerto de 80% na identificação dos alunos, considerando diferentes ângulos de rosto e variações na iluminação do ambiente. Portanto este resultado demonstra a eficácia e a capacidade do modelo utilizado em lidar com situações e adversidades reais de captura de imagens.

## 6. Considerações Finais

Ao longo deste trabalho, foram apresentados os métodos e etapas de desenvolvimento de um aplicativo capaz de realizar o processo de reconhecimento facial de forma nativa em dispositivos móveis, visando obter a melhor performance possível. A tecnologia empregada no reconhecimento facial está em constante crescimento e possui uma ampla gama de aplicações. Recentemente, devido à pandemia enfrentada nos últimos anos, houve um impulso exponencial no uso dessa tecnologia para diversos fins.

Durante o desenvolvimento de todo o estudo foram encontradas algumas dificuldades no processo de compreender o funcionamento das redes neurais e como se dá todo o processo de detecção e reconhecimento facial, que foram suplantados por meio de estudos teóricos e projetos práticos utilizando as tecnologias.

Com base nos estudos e resultados obtidos, respeitando as etapas da engenharia de software e resultados positivos nos testes de qualidades realizados com finalidade de validação, o objetivo do estudo foi alcançado. Fato que demonstra o potencial dessa tecnologia em fornecer soluções inovadoras e eficazes.

Espera-se que este estudo contribua no avanço do conhecimento nesta área específica do reconhecimento facial para aplicativos móveis e que essa pesquisa estimule e atraia mais pessoas interessadas em realizar pesquisas e desenvolvimento neste campo promissor. O aplicativo desenvolvido durante este estudo pode servir como uma base sólida para futuras melhorias e desenvolvimentos, tais como a automatização de todo o processo, permitindo o uso cada vez mais amplo dessa tecnologia.

## 7. Referências

- ADAMOVIĆ, S. et al. (2020) An efficient novel approach for iris recognition based on stylometric features and machine learning techniques. *Future Generation Computer Systems*. v. 107. p. 144–157.
- ALEX, N. S. et al. (2012) Pose invariant hand biometry using multiple databases. *Procedia Engineering*. International conference on modelling optimization and computing. v. 38. p. 3579–3586.
- Matheus Alberto. (2022) Flutter: o que é e tudo sobre o framework. Disponível em: <<https://www.alura.com.br/artigos/flutter>>. Acesso em: 19 de maio de 2023.
- Sheng Chen. MobileFaceNet. (2018) Disponível em: <<https://arxiv.org/abs/1804.07573>>. Acesso em: 19 de maio de 2023.
- BECK, C. A. M. R. et al. (2021) Os (ab)usos da tecnologia de reconhecimento facial na segurança pública e na prestação de serviços a partir da pandemia de covid-19. *Revista Pensamento Jurídico*. Online. São Paulo.
- COSTA, R.S., OLIVEIRA, S.R. (2019) O uso de tecnologias de reconhecimento facial em sistemas de vigilância e suas implicações no direito à privacidade. *Revista de Direito, Governança e Novas Tecnologias*. p. 01 – 21. Belém.
- CUSTÓDIO, K.V.A. (2007) Estudo do uso de biometria para autenticação em terminais de autoatendimento. Dissertação. Graduação em Engenharia da Computação. Faculdade de Ciências Exatas e Tecnológicas. Brasília.
- GOOGLE DEVELOPERS. MLKIT (2023) Disponível em:

- <<https://developers.google.com/ml-kit/vision/face-detection?hl=pt-br>>. Acesso em: 19 de maio de 2023.
- DIÁRIO-CATARINENSE, J. (2020) Depois de Florianópolis, governo quer reconhecimento facial em todos os aeroportos do país.
- FIREBASE. Cloud Firestore (2023) Disponível em: <<https://firebase.google.com/docs/firestore?hl=pt-br>>. Acesso em 19 de maio de 2023.
- LI, S. Z.; JAIN, A. K. (2011) Handbook of Face Recognition. London: Springer. E-book.
- LOPES, E.C. (2016) Detecção de faces e características faciais. Dissertação: Pós-Graduação em Ciência da Computação. Pontifícia Universidade Católica do Rio Grande do Sul. Porto Alegre.
- ORVALHO, V. (2019) Reconhecimento facial. Revista Ciência Elementar. V7(4):073. Porto.
- PHILLIPS, P. J.; MCCABE, R. M.; CHELLAPPA, R. (1998) Biometric image processing and recognition. In: European signal processing conference. p. 1–8.
- PUBDEV. (2019) Sqlite. Disponível em: <<https://pub.dev/packages/sqlite>>. Acesso em 19 de maio de 2023.
- SILVA, A. L.; CINTRA, M. E. (2015) Reconhecimento de padrões faciais: Um estudo. In: Encontro Nacional de Inteligência Artificial e Computacional. Proceedings ENIAC.
- SOUZA, R. R. (2019) Sobre a ética humana e a ética dos algoritmos. Horizonte presente: Debates de tecnologia e sociedade. In: Magrani, Eduardo. (Org.) p. 577-586. Rio de Janeiro.
- Filipe Garret. (2021) O que é Figma? Quatro perguntas sobre como usar o site. Disponível em: <<https://www.techtudo.com.br/listas/2021/06/o-que-e-figma-quatro-perguntas-sobre-como-usar-o-site.ghtml>>. Acesso em: 19 de maio de 2023.
- ICMC. (2023) Redes neurais. Disponível em: <<https://sites.icmc.usp.br/andre/research/neural/>>. Acesso em: 22 de maio de 2023.
- WIKIPEDIA. (2015) TensorFlow. Disponível em: <<https://en.wikipedia.org/w/index.php?oldid=802350135>>. Acesso em: 19 de maio de 2023.
- YANG, M. H. et al. (2002) Detecting faces in images: a survey. Ieee Transactions on Pattern Analysis and Machine Intelligence. p. 34-58.