**UNIVERSIDADE PAULISTA**

**Gabriel Soares Araújo**

**Lucas Amorim de Carvalho**

**Lucca Kazlauscas Zaccaria**

**Matheus Lopes de Jesus**

**CHECAGEM DE PRESENÇA POR RECONHECIMENTO FACIAL**

**São Paulo**

**2020**

**Gabriel Soares Araújo**

**Lucas Amorim de Carvalho**

**Lucca Kazlauscas Zaccaria**

**Matheus Lopes de Jesus**

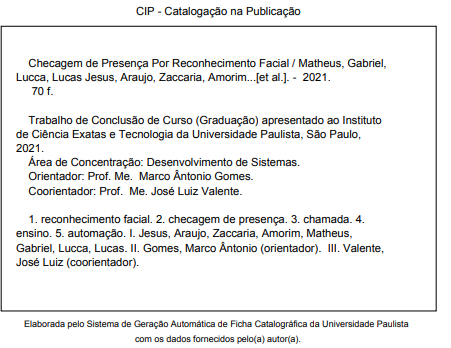
**CHECAGEM DE PRESENÇA POR RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso apresentado à UNIP – Universidade Paulista para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Marco Antônio

**São Paulo**

**2020**



**Gabriel Soares Araújo – N1645C0**

**Lucas Amorim de Carvalho – N164AB9**

**Lucca Kazlauscas Zaccaria – D508AE7**

**Matheus Lopes de Jesus – N145986**

**CHECAGEM DE PRESENÇA POR RECONHECIMENTO FACIAL**

Trabalho de Conclusão de Curso apresentado à UNIP – Universidade Paulista para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Marco Antônio

Aprovado em:

Banca Examinadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

Banca Examinadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

Banca Examinadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

Banca Examinadora

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_/\_\_\_

**RESUMO**

A educação é um pilar muito importante para qualquer nação, mas precisa ser olhado com mais atenção em países em desenvolvimento, como o Brasil, pois para que haja evolução e crescimento, precisa-se de conhecimento. O objetivo deste trabalho é através do uso de inteligência artificial, automatizar uma das atividades mais tradicionais do ensino, a checagem de presença. Para o desenvolvimento desse projeto foram feitas, pesquisa de campo e aplicação de questionários, para que pudéssemos ter o melhor entendimento do problema para criar uma solução que atenda às necessidades dos usuários. Com essa solução, esperamos alcançar um impacto positivo na absorção do conhecimento passado em sala de aula.

Palavras-chaves: Educação, Inteligência Artificial, Automação.

**ABSTRACT**

Education is a very important pillar for any nation, but it needs to be looked at more carefully in developing countries, like Brazil, because for there to be evolution and growth, knowledge is needed. The objective of this work is, through the use of artificial intelligence, to automate one of the most traditional teaching activities, the presence check. For the development of this project, field research and questionnaires were made, so that we could have the best understanding of the problem to create a solution that meets the needs of users. With this solution, we hope to achieve a positive impact on the absorption of knowledge passed in the classroom.

Key-words: Education, Artificial Intelligence, Automation

|  |
| --- |
| **LISTA DE ILUSTRAÇÕES**  Figura 1 – Estrutura da Equipe.........................................................................10  Figura 2 – Pergunta 1.......................................................................................12  Figura 3 – Pergunta 2.......................................................................................13  Figura 4 – Pergunta 3.......................................................................................14  Figura 5 – Pergunta 4.......................................................................................14  Figura 6 – Pergunta 5.......................................................................................15  Figura 7 – Pergunta 6.......................................................................................15  Figura 8 – Diagrama Caso de Uso...................................................................21  Figura 9 – Placa ESP32 CAM..........................................................................28  Figura 10 – Arquitetura da Solução Definida...................................................30  Figura 11 – Tela: Login....................................................................................32  Figura 12 – Tela: Visualizar Presenças............................................................33  Figura 13 – Tela: Controle de Aula...................................................................33  Figura 14 – Tela: Cadastro de Usuário.............................................................34  Figura 15 – Tela: Alteração de Presença..........................................................34  Figura 16 – Tela: Visualizar Alunos...................................................................35  Figura 17 – Tela: Edição de Usuário.................................................................35  Figura 18 – Diagrama de Entidade Relacionamento.........................................36  Figura 19 – Modelagem Relacional...................................................................37  Figura 20 – Diagrama de Caso de Uso.............................................................43  Figura 21 – Diagrama de Classes.....................................................................45  Figura 22 – Diagrama de Sequência.................................................................46  Figura 23 – Cronograma Parte 1.......................................................................47  Figura 24 – Cronograma Parte 2.......................................................................48 |

|  |
| --- |
| **LISTA DE TABELAS**  Tabela 1 – Primeira Pergunta para Professores...............................................18  Tabela 2 – Segunda Pergunta para Professores..............................................18  Tabela 3 – Terceira Pergunta para Professores...............................................18  Tabela 4 – Quarta Pergunta para Professores.................................................19  Tabela 5 – Quinta Pergunta para Professores..................................................19  Tabela 6 – Pergunta para Alunos......................................................................19  Tabela 7 – Requisito Secretaria(o)....................................................................23  Tabela 8 – Requisito Sistema............................................................................23  Tabela 9 – Requisito Professor(a).....................................................................23  Tabela 10 – Requisito Aluno(a)..........................................................................23  Tabela 11 – Modelagem Física: Aluno\_TB........................................................38  Tabela 12 – Modelagem Física: Usuario\_TB.....................................................39  Tabela 13 – Modelagem Física: Funcionario\_TB..............................................39  Tabela 14 – Modelagem Física: Aluno\_TB\_has\_Aulas\_TB..............................39  Tabela 15 – Modelagem Física: Aulas\_TB........................................................39  Tabela 16 – Modelagem Física: Salas\_TB........................................................40  Tabela 17 – Dicionário de Dados Aluno\_TB......................................................40  Tabela 18 – Dicionário de Dados Salas\_TB......................................................40  Tabela 19 – Dicionário de Dados Aluno\_TB\_has\_Aulas\_TB.............................41  Tabela 20 – Dicionário de Dados Usuario\_TB...................................................41  Tabela 21 – Dicionário de Dados Funcionario\_TB............................................42  Tabela 22 – Dicionário de Dados Aulas\_TB......................................................42  Tabela 23 – Narrativa do Caso de Uso..............................................................45  Tabela 24 - Levantamento Bibliográfico “Desenvolvimento de um protótipo  de software de reconhecimento facial: estudo do sistema presente”................50  Tabela 25 – Levantamento Bibliográfico “Sistemas de Reconhecimento  Facial ”...............................................................................................................51  Tabela 26 – Levantamento Bibliográfico “Estudo sobre métodos de  reconhecimento facial em fotografias digitais”..................................................52 |

**SUMÁRIO**

[1 INTRODUÇÃO 11](#_Toc62169224)

[1.1 JUSTIFICATIVA 12](#_Toc62169225)

[1.2 OBJETIVOS 14](#_Toc62169226)

[1.2.1 Objetivos gerais 14](#_Toc62169227)

[1.2.2 Objetivos específicos 14](#_Toc62169228)

[1.3 ESTRUTURA DA EQUIPE 15](#_Toc62169229)

[2 REFERENCIAL TEÓRICO 16](#_Toc62169230)

[3 MÉTODOS E MATERIAIS 17](#_Toc62169231)

[3.1 Levantamento de Requisitos 17](#_Toc62169232)

[3.1.1 Técnicas de levantamento de requisitos 17](#_Toc62169233)

[3.1.2 Questionário 18](#_Toc62169234)

[3.1.3 Entrevista 22](#_Toc62169235)

[3.2 Análise de Requisitos 25](#_Toc62169236)

[3.2.1 análise do problema 25](#_Toc62169237)

[3.2.2 avaliação e síntese 25](#_Toc62169238)

[3.2.3 modelagem 26](#_Toc62169239)

[3.2.4 especificação dos requisitos e revisão 28](#_Toc62169240)

[3.3 Estudo de Viabilidade 29](#_Toc62169241)

[3.3.1 Viabilidade financeira 29](#_Toc62169242)

[3.3.2 módulo de captura de imagem 29](#_Toc62169243)

[3.3.3 servidor para sistema de processamento e aplicação de gerenciamento 30](#_Toc62169244)

[3.3.4 custo de Implementação do projeto 30](#_Toc62169245)

[3.4 Viabilidade técnica 31](#_Toc62169246)

[3.5 Viabilidade legal 32](#_Toc62169247)

[3.6 Projeto de Software 33](#_Toc62169248)

[3.6.1 Módulo para Captura de Imagens 33](#_Toc62169249)

[3.6.2 Servidor para Reconhecimento Facial 33](#_Toc62169250)

[3.6.3 Infraestrutura Conteinerizada 33](#_Toc62169251)

[3.6.4 Aplicação WEB 33](#_Toc62169252)

[3.6.5 Base para Armazenamento de Dados 34](#_Toc62169253)

[3.6.6 Arquitetura da Solução 34](#_Toc62169254)

[3.7 Projeto de Interface Homem Maquina 36](#_Toc62169255)

[3.8 Definição da Modelagem de Dados 40](#_Toc62169256)

[3.8.1 Diagrama de entidade relacionamento (DER): 40](#_Toc62169257)

[3.8.2 Modelagem relacional 41](#_Toc62169258)

[3.8.3 Modelagem física (SQL) 41](#_Toc62169259)

[3.8.4 Dicionário de dados: 43](#_Toc62169260)

[3.9 DEFINIÇÃO DA MODELAGEM DE SISTEMAS 47](#_Toc62169261)

[3.9.1 Diagrama de caso de uso 47](#_Toc62169262)

[3.9.1.1 narrativa de caso de uso 48](#_Toc62169263)

[3.9.2 Diagrama de classes 49](#_Toc62169264)

[3.9.3 Diagrama de sequência 50](#_Toc62169265)

[3.10 Cronograma 51](#_Toc62169266)

[4 RESULTADOS 53](#_Toc62169267)

[5 CONCLUSÃO 54](#_Toc62169268)

[6 TRABALHOS FUTUROS 55](#_Toc62169269)

[7 REFERÊNCIA BIBLIOGRÁFICA 56](#_Toc62169270)

[7.1 LEVANTAMENTO BIBLIOGRÁFICO 56](#_Toc62169271)

[7.2 Referências Bibliográficas 59](#_Toc62169272)

[8 ANEXOS 61](#_Toc62169273)

# 1 INTRODUÇÃO

Quando pensamos em processos e atividades envolvendo escolas e o sistema de ensino em geral facilmente enxergamos pontos de melhoria, em especial no meio de ensino superior, ambiente em que a alta eficiência e produtividade são cobradas e postas em prática. Um desses pontos de melhoria definitivamente são as tarefas burocráticas, que tomam parte do tempo de ensino da aula, diminuindo, e muitas vezes se intercalando com o tempo produtivo, afetando o aprendizado.

Hoje no Brasil, diversos indicadores e índices apontam a estagnação, e em alguns casos a queda, da qualidade e nível do ensino, e uma das formas de melhorar esse cenário é adaptando e modificando os processos burocráticos que impactam o tempo produtivo de aula, em específico o processo de checagem de presença dos alunos (comumente denominado “chamada”), geralmente feito de forma manual, arcaica, e custando parte do tempo produtivo de aula.

Neste trabalho buscamos desenvolver um módulo que possa de forma eficiente e automatizada executar a checagem de presença dos alunos, dessa forma aumentando o tempo produtivo de aula, e assim, aplicando a tecnologia para melhorar um dos processos burocráticos utilizados no sistema de ensino.

Visando que o módulo elaborado seja viável, tanto financeiramente para as instituições de ensino, quanto tecnicamente para o desenvolvimento e escala da solução, utilizamos um microcontrolador de baixo custo comparado as demais opções no mercado, além de utilizarmos uma tecnologia de conteinerização contribuindo para a portabilidade do sistema como um todo, o que também viabiliza a sua aplicação em um cenário prático real.

Neste trabalho objetivamos impactar positivamente a eficiência e a qualidade do ensino, aplicando a tecnologia e solução descritas aqui para modernizar um processo comum nas instituições de ensino, a checagem de presença dos alunos, seja à nível de ensino básico, médio ou superior.

## 1.2 Contextualização

## 1.3 Motivação

## 1.4 Justificativa

De acordo com o Banco Mundial, no Brasil, 34% do tempo de aula é gasto com atividades burocráticas, sendo elas diversas, desde realizar chamadas, até entregar e receber atividades e tarefas extraclasse, sobrando apenas 66% do tempo para ensino e aprendizagem propriamente ditos. Se compararmos esse contexto com a média dos demais países da OCDE (Organização para Cooperação e Desenvolvimento Econômico), vemos nestes países um cenário melhor, em que 85% do tempo, em média, é aproveitado para a absorção de conhecimento. Nestes mesmos países, considerando uma aula de 50 minutos, temos 42 minutos de aprendizagem, enquanto no Brasil, temos 33 minutos de aprendizagem para os mesmos 50 minutos de aula.

Considerando os dados apresentados, o contexto brasileiro, e o cenário de 200 dias letivos por ano com 5 aulas por dia, podemos estimar uma perda de 17000 minutos/ano, ou 283 horas e 20 minutos por ano, que não são utilizados para aprendizado.

A importância do aproveitamento e da eficiência do tempo de aula torna-se evidente quando analisamos a estagnação da qualidade de ensino no Brasil comparado a outros países.

Trazer para as pesquisas e discussões o objetivo de melhorar o aproveitamento do tempo de aula no Brasil pode ser um passo decisivo para alavancar a qualidade de ensino, o que pode causar impactos diretos e indiretos em diversas questões sociais, como na qualidade da Pesquisa e Desenvolvimento nacionais além dos índices de desemprego, visto que mais pessoas seriam capacitadas de forma eficiente. Por outro lado, negar o espaço para melhorias no aproveitamento do tempo de aula, ignorando essa discussão, pode significar perder uma forma potente de melhorar o sistema de ensino como um todo, contribuindo para a inércia e estagnação da qualidade de ensino.

Além do tema deste trabalho trazer um aspecto prático muito relevante, podendo causar impactos positivos diretos na sociedade, também apresenta importância para o meio acadêmico. Nesse contexto, a produção de pesquisas e a aplicação de tecnologias e ferramentas para o sistema de ensino indica o passo inicial para um processo de transformação em um modelo de ensino antigo, antiquado e pouco modernizado, que pode e deve ser revisitado e adaptado para os dias de hoje.

## 1.5 Objetivos

### 1.5.1 Objetivos gerais

Esse projeto tem como objetivo automatizar o processo de checagem de presença de alunos em salas de aula utilizando reconhecimento facial.

### 1.5.2 Objetivos específicos

Buscamos alcançar o objetivo acima, através da elaboração de um módulo para captura das imagens dos alunos em sala de aula, da definição de uma solução distribuída, da implementação de um algoritmo que realiza o reconhecimento facial, e do desenvolvimento de um algoritmo para a validação da presença do aluno em sala.

## 1.6 Metodologias de Pesquisa

1.6.1 Levantamento de Requisitos

Para alcançarmos a proposta desse projeto, fizemos o levantamento de requisitos, que é um processo fundamental para o desenvolvimento de um projeto, pois através dele conseguimos garantir a correta compreensão e identificação das necessidades e expectativas do cliente. Realizamos este processo através de estudo de campo e pesquisas aplicando as técnicas de levantamento de requisitos.

### 1.6.1.1 Técnicas de levantamento de requisitos

As técnicas de levantamento de requisitos visam auxiliar no desenvolvimento do projeto, para que ele seja feito de forma mais assertiva e eficiente. As técnicas utilizadas nesse projeto foram:

1. Levantamento de Requisitos Orientado à Ponto de Vista;
2. Questionário;
3. Entrevista.

#### 1.6.1.1 Levantamento de requisitos à ponto de vista

##### 1.6.1.1.1 usuários do sistema

Para o Levantamento de Requisitos Orientado aos Pontos de Vista de cada parte envolvida no sistema, e para o desenvolvimento da solução como um todo, foram consideradas as seguintes partes, baseadas nas suas interações com o sistema:

1. Funcionários da Secretaria: Responsáveis pelo cadastro de alunos e funcionários no sistema da instituição, e cadastro de aulas para os professores;
2. Professores: Responsáveis pelo começo efetivo da aula e pela checagem da presença do aluno na aula;
3. Alunos: Pessoa matriculada na instituição e com o dever de presença na aula em que estiver cadastrado.

##### 1.6.1.2 interações por ponto de vista

Dado o grupo de usuários do sistema, temos as seguintes atividades executadas por cada um, e suas interações com a solução proposta:

* + - * Funcionários da Secretaria:
        + Cadastrar alunos.
        + Alterar dados dos alunos.
        + Cadastrar funcionários.
        + Cadastrar aulas para professores.
        + Alterar dados de funcionários.
        + Alterar dados de aulas.
      * Professores:
        + Controlar o início das aulas.
        + Alterar presença dos alunos.
      * Alunos:
        + Visualizar sua lista de presenças.

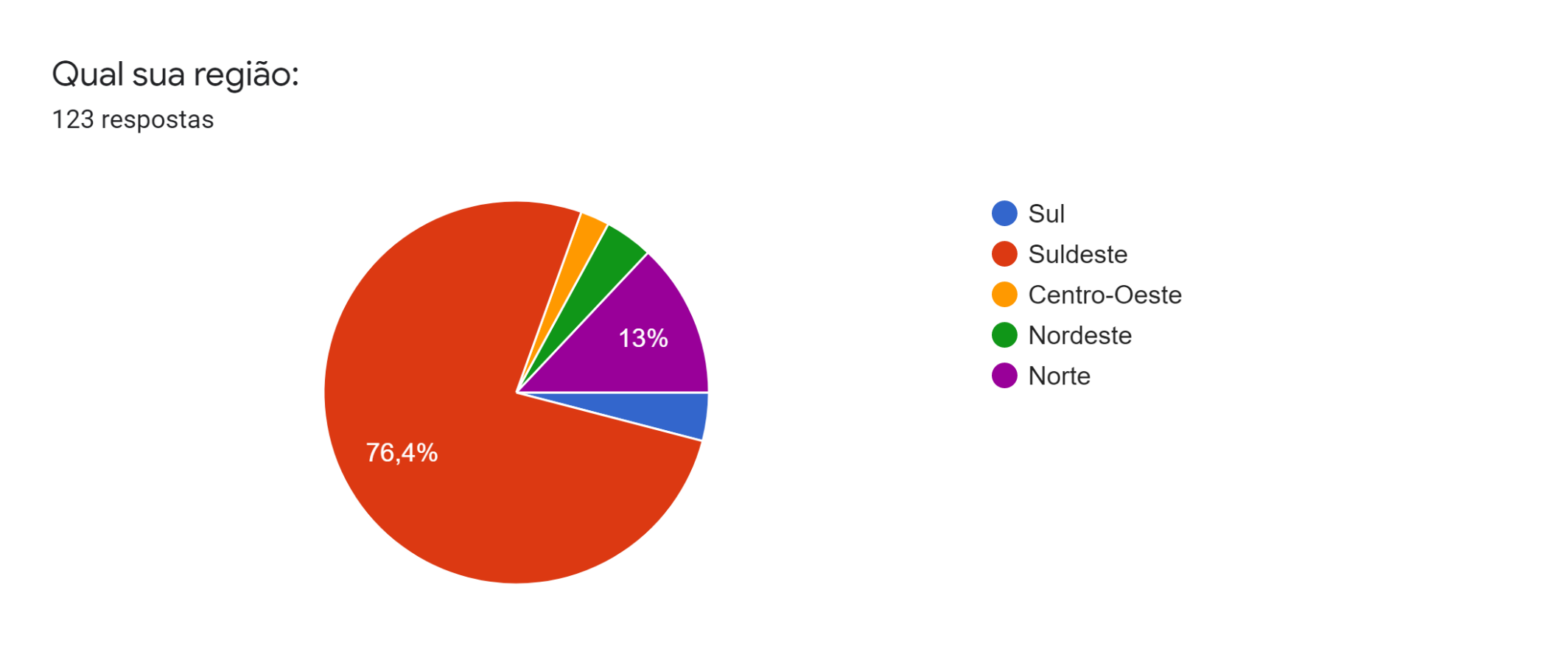
### 1.6.1.2 Questionário

Como uma das formas de alinhar as necessidades e expectativas sobre a solução, e capturar as opiniões das partes envolvidas sobre o tema da checagem de presença feita em sala de aula de forma não automatizada, utilizamos um questionário:

#### 1.6.1.2.1 perguntas

1. Em qual região você atua?

Figura 2 – Pergunta 1



Fonte: Google Formulário, 2020.

1. Em sua opinião, atividades burocráticas em sala como: checagem de presença oral (“chamada” oral), entrega de trabalhos, apagar/escrever na lousa, entre outras, influencia no tempo produtivo de aula?

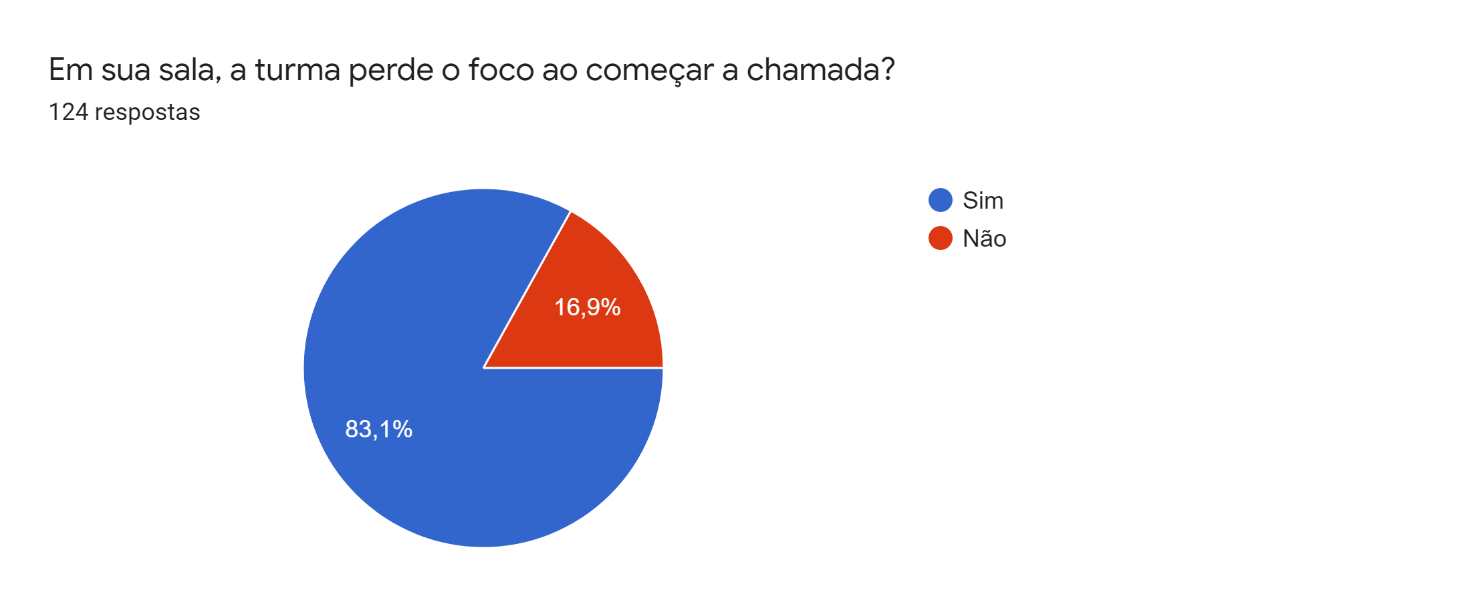
Figura 3 – Pergunta 2



Fonte: Google Formulário, 2020.

1. Na sua experiência, há perda de foco dos alunos ao começar a checagem de presença?

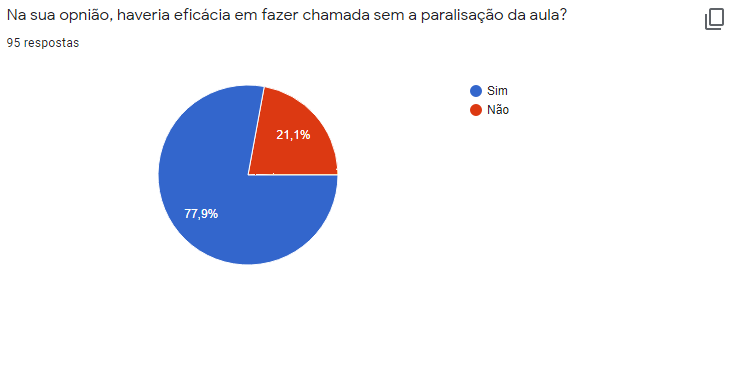
Figura 4 – Pergunta 3



Fonte: Google Formulário, 2020.

1. Na sua opinião, haveria ganho em fazer checagem de presença sem a paralisação da aula?

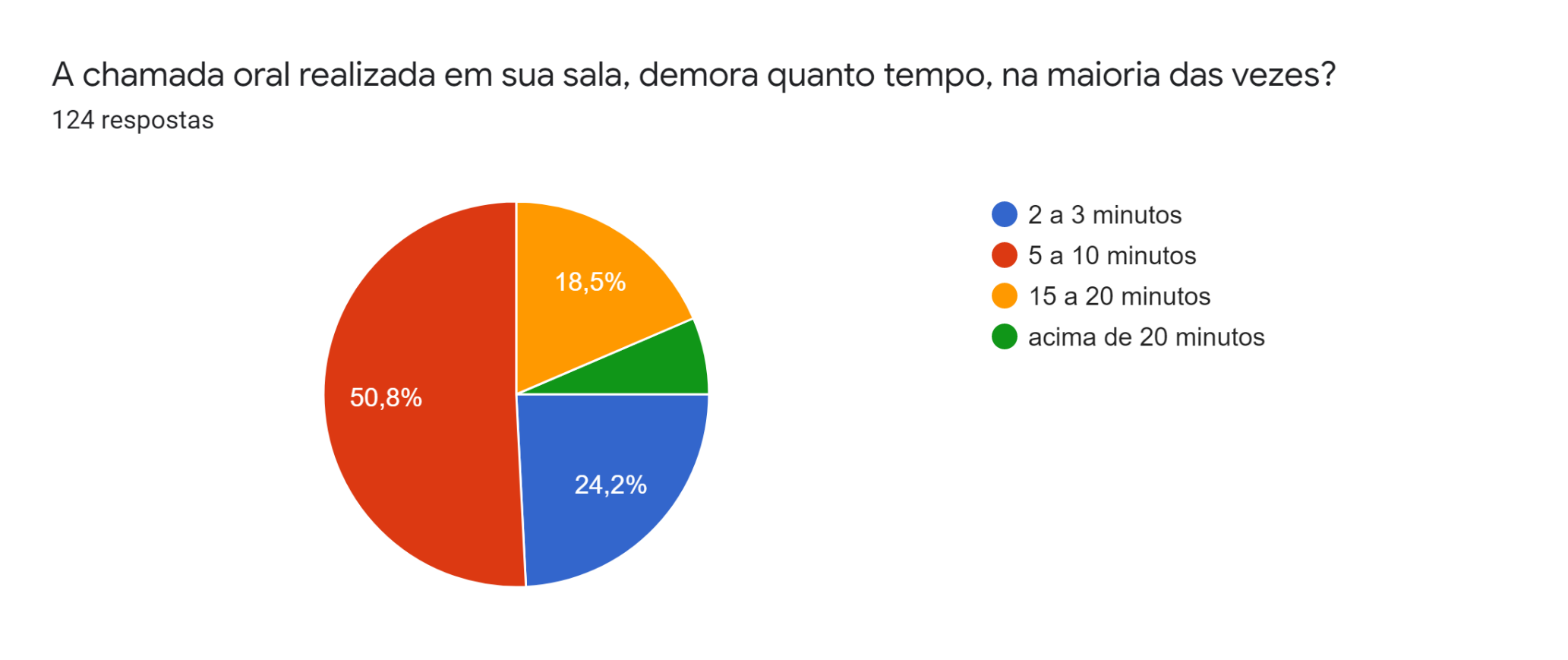
Figura 5 – Pergunta 4



Fonte: Google Formulário, 2020.

5. A checagem de presença oral realizada em suas aulas demora quanto tempo, na maioria das vezes?

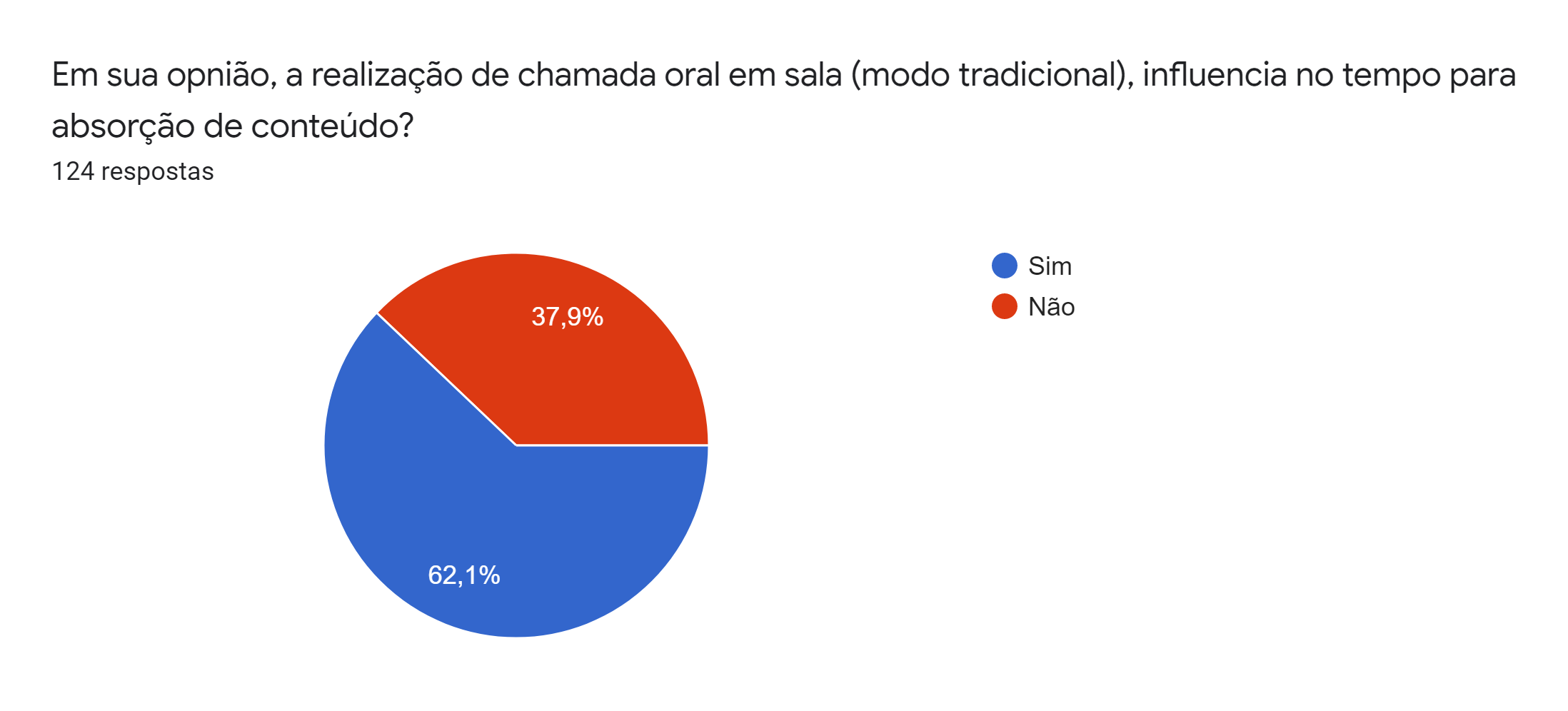
Figura 6 – Pergunta 5



Fonte: Google Formulário, 2020.

1. Em sua opinião, a realização da checagem de presença oral em sala (método comum), influencia no tempo útil para absorção de conteúdo?

Figura 7 – Pergunta 6



Fonte: Google Formulário, 2020.

#### 1.6.1.2.2 conclusões do questionário

Com o resultado do questionário podemos concluir que a maior parte dos entrevistados concorda que as tarefas burocráticas causam um impacto considerável no tempo de aula, variando em média de 5 à 20 minutos de impacto, o que vai de acordo com os dados da OCDE de uma média de 17 minutos de impacto causado por tarefas burocráticas a cada aula de 50 minutos.

Podemos reparar também, que a maior parte das pessoas indicou que há uma perda de foco na aula após iniciar a checagem de presença, e que a maior parte dos dados coletados são de pessoas da região sudoeste do país.

1.6.2 Análise de Requisitos

### 1.6.2.1 análise do problema

Através de entrevistas, questionários e pesquisas pôde se encontrar as dores que os processos burocráticos podem causar. Os resultados gerados através do levantamento de requisitos, possibilitou que fosse analisado o contexto em que este projeto está inserido para tornar possível a sua aplicação.

Ao realizar o processo de levantamento de requisitos, foi observado que as atividades burocráticas realizadas em sala, causam um impacto significativo no tempo de aula, podendo chegar a vinte minutos comprometidos. Além da realização dessas atividades, outro ponto que foi destacado, é a perda do foco dos alunos.

### 1.6.2.2 avaliação e síntese

Com as conclusões obtidas no tópico acima, foi entendido que as atividades burocráticas tem influência na absorção de conteúdo em sala. Pensando em contribuir com a evolução do sistema de ensino brasileiro, foi pensado em uma solução que pudesse abordar alguma das atividades que mais gera impacto no tempo de aula, a checagem de presença.

Através do uso da inteligência artificial e o uso de microcontrolador, a solução proposta é de automatizar o processo de checagem de presença em salas de aula.

Para a realização dessa solução, será montado um módulo que consiste em um microcontrolador e uma câmera, para que haja a captação de imagens dos alunos presentes em sala. Além do módulo, terá um sistema para o gerenciamento das presenças, onde alunos podem visualizar suas presenças e os professores alterá-las, se necessário. Através de requisições HTTP(Hypertext Transfer Protocol), o sistema se comunica com o módulo e recebe as fotos, ao receber essas imagens, realiza o processo de reconhecimento facial e valida a presença ou não.

### 1.6.2.3 modelagem

Segue abaixo diagramas para elucidar o funcionamento da solução:

O diagrama de caso de uso abaixo, ilustra as funções que o sistema realizará e quem pode realiza-las

Figura 8 – Diagrama Caso de Uso

Diagrama

Descrição gerada automaticamente

Fonte: Autores, 2020

### 1.6.2.4 especificação dos requisitos e revisão

**Tabela 7 – Requisitos Secretaria(o)**

|  |
| --- |
| **Secretaria(o)** |
| RF01: Cadastrar usuários |
| RNF01: O usuário pode ser do tipo aluno ou professor |
| RF02: Editar usuários |
| RNF02: Pode editar informações de alunos e professores |

Fonte: Autores, 2020.

**Tabela 8 – Requisitos Sistema**

|  |
| --- |
| **Sistema** |
| RF01: Controlar presença |
| RF02: Reconhecer faces |

Fonte: Autores, 2020.

**Tabela 9 – Requisitos Professor**

|  |
| --- |
| **Professor(a)** |
| RF01: Controlar aulas |
| RNF01: O professor pode cancelar aula |
| RF02: Inicia aula |
| RF03: Alterar presença |

Fonte: Autores, 2020.

**Tabela 10 – Requisitos Aluno(a)**

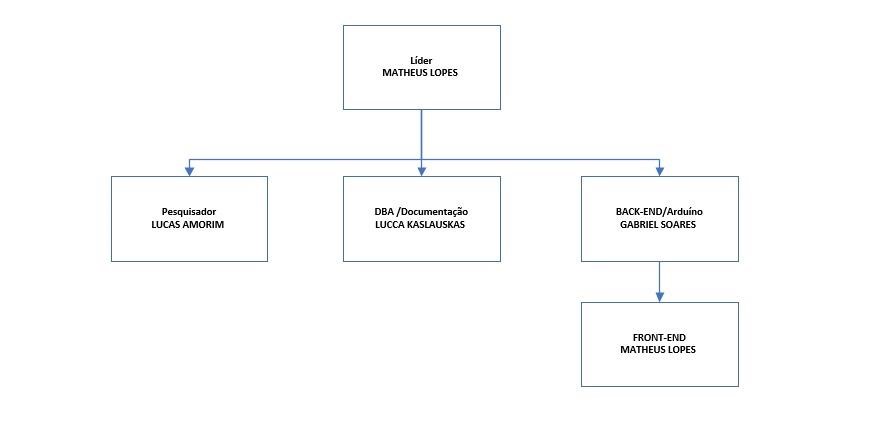
|  |
| --- |
| **Aluno(a)** |
| RF01: Visualizar presença |
| RF02: Visualizar aulas |

Fonte: Autores, 2020.

## 1.7 Estrutura do Trabalho

Para obtermos um resultado satisfatório na entrega do projeto, dividimos a equipe, onde cada integrante irá atuar em uma área específica. Segue abaixo uma ilustração da estrutura.

Figura 1 – Estrutura da Equipe



Fonte: Autores, 2020.

Matheus Lopes, na posição de líder, gerencia as atividades, prazos e qualidade nas entregas, e realiza a montagem e programação do Hardware*.*

Gabriel Soares, analista desenvolvedor, realiza o desenvolvimento *back-end* do software, e também é responsável pela administração da base de dados.

Lucca Kazlauskas, como analista de documentação, desenvolve a documentação e registro do projeto e auxilia no desenvolvimento front-end.

Lucas Amorim atua como pesquisador. Responsável por realizar pesquisas e o desenvolvimento do projeto escrito, auxiliando na montagem do hardware.

# 2 REFERENCIAL TEÓRICO

## 2.1 Projeto de Software

A solução apresentada neste projeto tem como característica uma arquitetura distribuída, podendo ser segregada nas seguintes responsabilidades:

* Módulo para Captura de Imagens;
* Servidor para Reconhecimento Facial;
* Aplicação WEB;
* Base para Armazenamento de Dados.

### 2.1.1 Módulo para Captura de Imagens

Para o módulo que possuí a responsabilidade de capturar as imagens dos presentes na sala, foi utilizado um microcontrolador chamado ESP32-CAM, um microcontrolador de baixo custo financeiro e baixo custo energético para o funcionamento.

A placa, produzida pela empresa chinesa *Espressif Systems,* é considerada uma SoC (do inglês “System on Chip”, ou “Sistema em um Chip”), ou seja, contém todos os recursos integrados na mesma placa, que possuí as funções Wi-Fi e Bluetooth, além da câmera OV2640, 512 KB de memória RAM, 32 Mb de memória Flash, e um processador ESP32-D0WD.

Para o desenvolvimento do software embarcado, foi utilizada a IDE Arduino como ambiente de desenvolvimento, com a linguagem de programação C.

O funcionamento do módulo é dado pela disponibilização de um módulo por sala de aula, tendo um *endpoint* na rede local para que, ao receber uma requisição HTTP neste *endpoint*, o módulo então capture uma imagem, e a devolva para o *client* da requisição. Sua responsabilidade é delimitada por esta ação: retornar a imagem da sala de aula quando solicitada para que o servidor realize o reconhecimento facial.

Figura 9 – Placa ESP32-CAM



Fonte: Auto Core Robotica, 2020.

### 2.1.2 Servidor para Reconhecimento Facial

Para o servidor que executará o processamento das imagens e o reconhecimento facial, utilizamos a linguagem de programação JavaScript com a plataforma de execução Node.js, que é a responsável por interpretar o código JavaScript no lado do servidor (*server-side*).

A responsabilidade do servidor é definida por duas partes, sendo a primeira, a função de intermediar as ações da *aplicação WEB* (Tópico 8.3), processando qualquer cadastro, atualização de registro, ou remoção feita na aplicação WEB por algum usuário, agindo como a camada de controle da arquitetura, baseando-se no padrão MVC (Model, View e Controller).

A segunda responsabilidade da aplicação do servidor, é a de executar o reconhecimento facial. Quando a aula for iniciada pelo professor (ação executada na aplicação WEB – Tópico 8.3), o servidor irá solicitar ao módulo a imagem, realizando uma requisição HTTP para o endpoint definido na rede local, e ao receber a imagem, fará o reconhecimento facial, checando se o aluno está presente na imagem feita. Este processo será feito diversas vezes durante a aula, para garantir que o aluno esteja em sala durante pelo menos 75% do tempo de aula, e após o final da aula, a aplicação do servidor irá registrar a presença para o aluno ou a falta dela.

### 2.1.3 Infraestrutura Conteinerizada

Para o desenvolvimento da solução, a viabilidade foi bastante considerada, e buscando aumentar a facilidade de implantação da solução, foi definido o uso da tecnologia de conteinerização chamada Docker.

Com o uso desta tecnologia, a aplicação do servidor ganha em portabilidade, pois se torna independente de sistema operacional e de hardware, desta forma, é criada a imagem da aplicação do servidor sendo executada com o Node.js, e essa imagem pode ser executada em qualquer hardware capaz de executar o Docker.

Com esta característica, a definição do servidor se torna muito versátil, podendo-se definir como servidor desde supercomputadores robustos até placas com recursos computacionais mais reduzidos, como por exemplo uma Raspberry Pi, adequando assim o servidor à necessidade computacional, evitando gastos desnecessários.

### 2.1.4 Aplicação WEB

A aplicação WEB desenvolvida tem o objetivo de apresentar uma interface gráfica intuitiva para que toda a solução seja administrada. Todas as interações dos usuários com a solução, com exceção das fotos tiradas pelo módulo, são dados através da aplicação WEB.

Na aplicação WEB o(a) Secretário(a) irá cadastrar o(a) Aluno(a) ou atualizar seus dados, o(a) Aluno(a) irá checar suas presenças e aulas, e o(a) Professor(a) irá iniciar ou cancelar suas aulas, além de definir o tempo de aula.

Esta aplicação foi desenvolvida utilizando como linguagem o TypeScript (superset da linguagem JavaScript) com o Framework Angular, facilitando o desenvolvimento do front-end com práticas como a componentização do front-end, e trabalhando com a aplicação do servidor aplicando o padrão *client-server*, utilizando o servidor como um conjunto de Restful APIs, ou seja, programas que vão ser manipulados e utilizados através dos métodos e requisições HTTP.

### 2.1.5 Base para Armazenamento de Dados

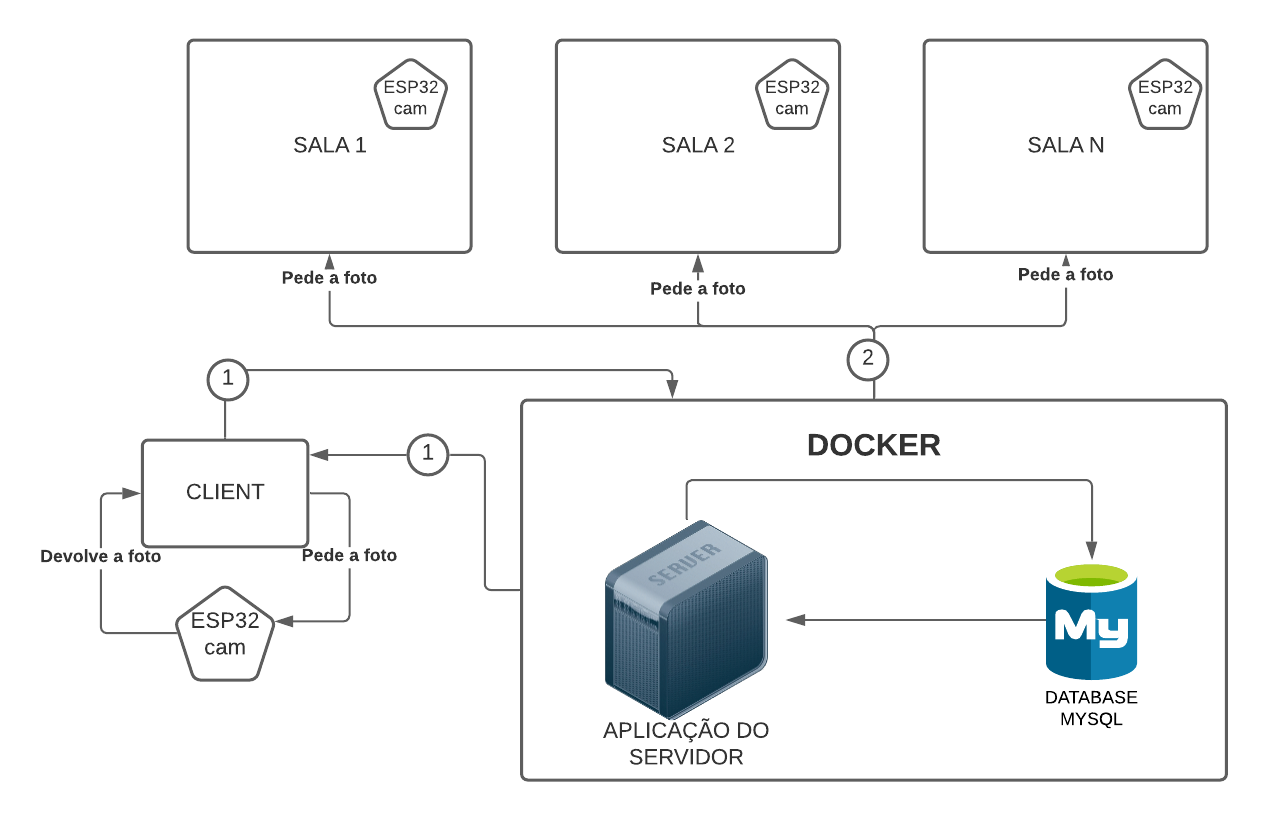
O armazenamento dos dados foi centralizado em uma única base de dados MySQL, da empresa Oracle, que utiliza o motor MariaDB. Nesse banco de dados serão armazenadas as informações referentes aos alunos, professores, secretários, aulas, e salas de aula, além das presenças registradas, e todo o fluxo de utilização desta base de dados é controlado e passa pela aplicação do servidor, nenhuma outra aplicação tendo acesso à base. As definições da modelagem da base são descritas em um tópico específico para isto ao longo deste trabalho.

É válido pontuar que, somente o armazenamento das imagens é feito de forma isolada, sem o uso da base de dados, salvando as imagens por tempo determinado diretamente no armazenamento de disco principal.

### 2.1.6 Arquitetura da Solução

A arquitetura da solução é definida pela imagem abaixo:

Figura 10 – Arquitetura da Solução Definida.



Fonte: Autores, 2020.

Na arquitetura definida, temos algumas constatações:

Um módulo é necessário por sala, assim tirando a foto para a checagem de presença de todas as aulas ministradas naquela sala;

Um módulo é necessário para que o(a) Secretário(a) possa realizar o cadastro do(a) Aluno(a);

A aplicação do servidor e a base de dados estão centralizados no mesmo servidor;

Ambos, tanto o servidor quanto os módulos devem estar em uma mesma rede.

Seguindo o modelo de arquitetura, temos dois fluxos principais na solução, **o cadastro**, e a **checagem de presença**.

No cadastro(*1*), o(a) Secretário(a) através da aplicação WEB faz o preenchimento dos dados do usuário, e utiliza o módulo que é ativado para tirar a primeira foto do aluno, enviando todos os dados para a aplicação do servidor armazenar na base centralizada.

Na checagem de presença(2), o(a) Professor(a) inicia a aula, e então o servidor calcula o intervalo em que as fotos serão tiradas para a checagem da presença, e utilizando este intervalo, requisita que o módulo presente naquela sala devolva uma imagem. Após isso, a aplicação do servidor faz o reconhecimento facial, utilizando as fotos tiradas no cadastro do aluno, e então registra as presenças, ou a falta delas.

A arquitetura da solução foi definida desta maneira para que fosse distribuída escalável, podendo suportar um número não definido de salas de aula e alunos, sendo assim uma solução viável tanto para instituições de ensino menores, quanto para as maiores instituições de ensino.

## 2.2 Estudo de Viabilidade

Após a completa compreensão sobre o domínio da solução e seus requisitos, foi executado um estudo preliminar de viabilidade, visando garantir que a solução como um todo seja acessível e aplicável para todas as partes envolvidas, aumentando a chance de sucesso do projeto. Foram analisadas as viabilidades do ponto de vista:

* Financeiro;
* Técnico;
* Legal;

### **2.2.1 Viabilidade financeira**

Para análise da viabilidade financeira, segregamos a solução nas seguintes partes:

* Módulo de captura de imagem;
* Servidor para sistema de processamento e aplicação de gerenciamento.

### **2.2.1.1 módulo de captura de imagem**

Para a produção de cada módulo de captura de imagem é utilizado:

* 1x ESP32-CAM (microcontrolador):



(<https://pt.aliexpress.com/item/4001212309479.html> acesso em 25/01/21 20:00)

O custo do módulo pode variar. Como fonte para esse dado, consultamos diversos sites de venda de eletrônicos, com isso constatamos que existem opções viáveis de importações, por exemplo da Inglaterra com o custo de R$54,50 através do site dx.com, ou da China com o custo de R$46,40 através do aliexpress.com. Além disso, dentre as opções nacionais temos as lojas virtuais wjcomponentes.com.br, mercadolivre.com.br/perfil/AGUIAR+BM, e americanas.com.br, por exemplo, com respectivamente o custo de R$54,99, R$49,80, e R$59,99. Temos assim, o custo médio nacional aproximado de R$54,93.

* 1x Conector fêmea P4:



(https://www.robocore.net/conector/conector-dc-p4-femea acesso em 25/01/21 20:00)

Para o conector da fonte, consultamos as lojas virtuais robocore.net, autocorerobotica.com.br, e eletrogate.com, todas nacionais e com o valor de venda do produto de, respectivamente, R$0,90, R$0,90, e R$1,00. Dessa forma, o custo médio aproximado do produto é de R$0,93.

* 1x Fonte 5V 1A bivolt:

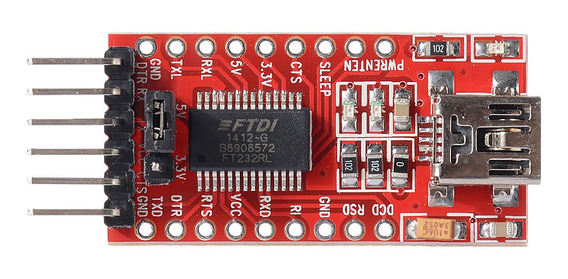


(<https://www.robocore.net/fonte/fonte-chaveada-5v-1a> acesso em 25/01/2021 20:00)

Consultamos o custo da fonte de alimentação utilizada no módulo nas lojas virtuais acessadas através de americanas.com.br, eletrogate.com, e submarino.com.br, com o respectivo preço de R$13,00, R$16,90, e R$13,00. O produto possui então, o custo médio aproximado de R$14,30.

Temos então que o custo de produção por módulo é de aproximadamente R$70,16.

O custo de produção indicado acima é aplicado por módulo produzido, mas para que a produção desses módulos seja possível, é necessário 1x Placa FTDI conversor Serial, com o custo médio aproximado de R$25,50, como indicado pelas lojas virtuais lojadahora.com.br (R$25,90) e submarino.com.br (R$25,11). Temos assim, como custo total de produção dos módulos de captura de imagem, o valor de: 25,50 reais + 70,16 reais/módulo. Sendo, portanto, 70,16 reais o preço de custo por unidade do módulo.



(https://arduxop.com/loja/produto/modulo-conversor-usb-serial-ftdi/ acesso em 25/01/2021 20:00)

### **2.2.1.2 servidor para sistema de processamento e aplicação de gerenciamento**

Para que a solução seja aplicada, é necessário um servidor na rede local que executará tanto o sistema de processamento das imagens e reconhecimento facial, quanto a aplicação WEB de gerenciamento do sistema.

Dependendo do cliente e/ou das partes interessadas, isso pode não representar um custo para a implantação da solução, visto que muitas instituições de ensino já possuem servidores disponíveis que poderiam ser utilizados para isso. Caso seja necessário para a implantação da solução a criação de servidores, estes podem ser providos de diversas formas, visto que a solução com o uso da técnica de conteinerização possuí portabilidade.

Abaixo temos o custo aproximado para que sejam providos os servidores em 2 cenários técnicos diferentes:

* Uso de uma placa Raspberry Pi 4 Modelo B 8GB:



(https://www.robocore.net/placa-raspberry-pi/raspberry-pi-4-8gb acesso em 25/01/2021 20:00)

Computador em placa única com capacidade computacional para executar a solução, sendo vendida pela robocore.net, revendedora oficial da Raspberry, ao valor de R$899,90. Esta opção seria mais indicada para instituições de ensino menores, com menor necessidade computacional para o processamento da presença dos alunos.

* Uso de serviço Amazon EC2 (t3a.xlarge sob demanda):

Com custo de $295,15/mês (dólares estadunidenses por mês de uso), segundo calculadora fornecida pela prestadora do serviço de cloud, considerando o uso de 75%/mês da máquina. Esta opção seria mais indicada para instituições de ensino maiores, com maior necessidade computacional para o processamento da presença dos alunos.

Vale ressaltar que o custo pode variar de acordo com a demanda de capacidade computacional. Quanto mais salas para serem processadas as checagens de presença, mais capacidade computacional será necessária.

### **2.2.1.3 custo de Implementação do projeto**

Para a implementação deste projeto em uma instituição de ensino, o custo final pode variar bastante de acordo com diversos fatores, como a instituição ter ou não um computador para ser utilizado como servidor, a quantidade de alunos por período, e até a frequência da coleta de imagens da sala de aula.

Para exemplificar o valor final da implementação, definimos como exemplo para o caso de uso, uma instituição de ensino que contenha 8 salas por período com uma média de 60 alunos por sala, totalizando por período 480 alunos, com uma frequência de coleta das imagens da sala de 10 minutos, para cada aula de 50 minutos.

Para o exemplo de instituição aqui definido, temos como custo para os 8 módulos (1 unidade por sala) de R$586,78. Para o servidor, indicamos o uso de um computador com as configurações especificadas nos testes de performance em anexo, uma opção para isso é o notebook Lenovo Ideapad S145, que conforme as lojas virtuais consultadas americanas.com.br (R$2789,07) e casasbahia.com.br (R$2399,00), tem o custo médio de R$2594,04.

Dessa forma, o custo final para a implementação do projeto em uma instituição de ensino com os padrões definidos acima é de: R$3180,82.

## 2.2.4 Viabilidade técnica

A elaboração da solução apresentada neste trabalho utiliza aplicações e ferramentas gratuitas, e em sua maioria de código aberto, como:

* Arduino IDE (para o desenvolvimento);
* VS Code (para o desenvolvimento);
* ngular (para o front-end);
* Node.js (para o back-end);
* Bibliotecas de código aberto (para o reconhecimento facial);
* MySQL (para a base de dados);
* Docker (para a conteinerização).

Além disso, para a elaboração do módulo e todo o hardware necessário os recursos já estão disponíveis, como detalhado na seção de viabilidade financeira e custo do protótipo.

O desenvolvimento dos programas que compõe a solução é de complexidade considerável por ser uma solução de arquitetura distribuída e exigindo atenção em muitos pontos diferentes do processo, desde o cadastro do aluno, à captura das imagens, e ao processamento da imagem. Apesar disso, a solução se apresenta como viável tecnicamente, exigindo para a implementação os seguintes conhecimentos:

* Redes e Comunicação HTTP;
* Linguagem C;
* Linguagem JavaScript;
* Html e CSS;
* Arquitetura embarcada;
* SQL e administração de bases de dados;
* E conhecimento básico de sistemas operacionais.

## 2.2.5 Viabilidade legal

Qualquer projeto de solução tecnológica deve levar em consideração a questão jurídica, visto que os sistemas intermediarão as interações humanas, que caso estejam ilegais ou irregulares, podem levar os desenvolvedores da solução a responderem como possíveis cúmplices de um crime, por exemplo.

Dado o propósito e contexto da solução proposta neste trabalho, deve-se atentar a Lei de Diretrizes e Bases da Educação Nacional (LDBEN), Lei 9394/96. Nela temos todas as definições de escopo nacional que regulamentam o sistema de ensino quanto a deveres, responsabilidades, e padrões a serem seguidos. Como por exemplo, o Título 4, Art. 13, Tópico V:

*Os docentes incumbir-se-ão de: ministrar os dias letivos e horas-aula estabelecidos, além de participar integralmente dos períodos dedicados ao planejamento, à avaliação e ao desenvolvimento profissional.* (Lei 9394/96, Título 4, Artigo 13, Tópico V, 1996)

A solução proposta deve colaborar para o correto cumprimento da lei citada, além de qualquer outra regulamentação que atue sobre o sistema de ensino. Incluindo o regulamento interno da instituição de ensino, o qual deve ser alinhado no processo de implantação da solução.

Em relação ao direito de imagem, apesar de a foto dos alunos ficar isolada nos servidores sem acesso de terceiros, nem a publicação ou uso das imagens para qualquer benefício que não seja a checagem da presença, também deve haver um alinhamento com a instituição de ensino, e a análise do contrato dos alunos com a instituição, visto que conforme o Artigo 20 do Código Civil de 2002:

*Salvo se autorizadas, ou se necessárias à administração da justiça ou à manutenção da ordem pública, a divulgação de escritos, a transmissão da palavra, ou a publicação, a exposição ou a utilização da imagem de uma pessoa poderão ser proibidas, a seu requerimento e sem prejuízo da indenização que couber, se lhe atingirem a honra, a boa fama ou a respeitabilidade, ou se se destinarem a fins comerciais.* (Código Civil, Artigo 20, 2002)

## 2.3 Definição de Modelagem de Sistemas

### 2.3.1 Diagrama de caso de uso

Figura 20 – Diagrama de Caso de Uso

Diagrama

Descrição gerada automaticamente

Fonte: Autores, 2020.

### 2.3.1.1 narrativa de caso de uso

**Tabela 23 – Narrativa Caso de Uso**

|  |
| --- |
| **Caso de uso:** Chamada por reconhecimento facial |
| **Atores:** Sistema, Secretaria(o), Professora(o), Aluna(o) |
| **Visão geral**: A Secretária(o) realiza o cadastro e edita informações dos alunos. O Professor(a) controla a aula, podendo inicia-la ou cancelá-la e pode alterar a presença dos alunos. O Sistema controla a presença dos alunos através do reconhecimento facial. O Aluno vizualiza a presença e as aulas. |

Fonte: Autores, 2020.

### 2.3.2 Diagrama de classes

Figura 21 – Diagrama de Classes

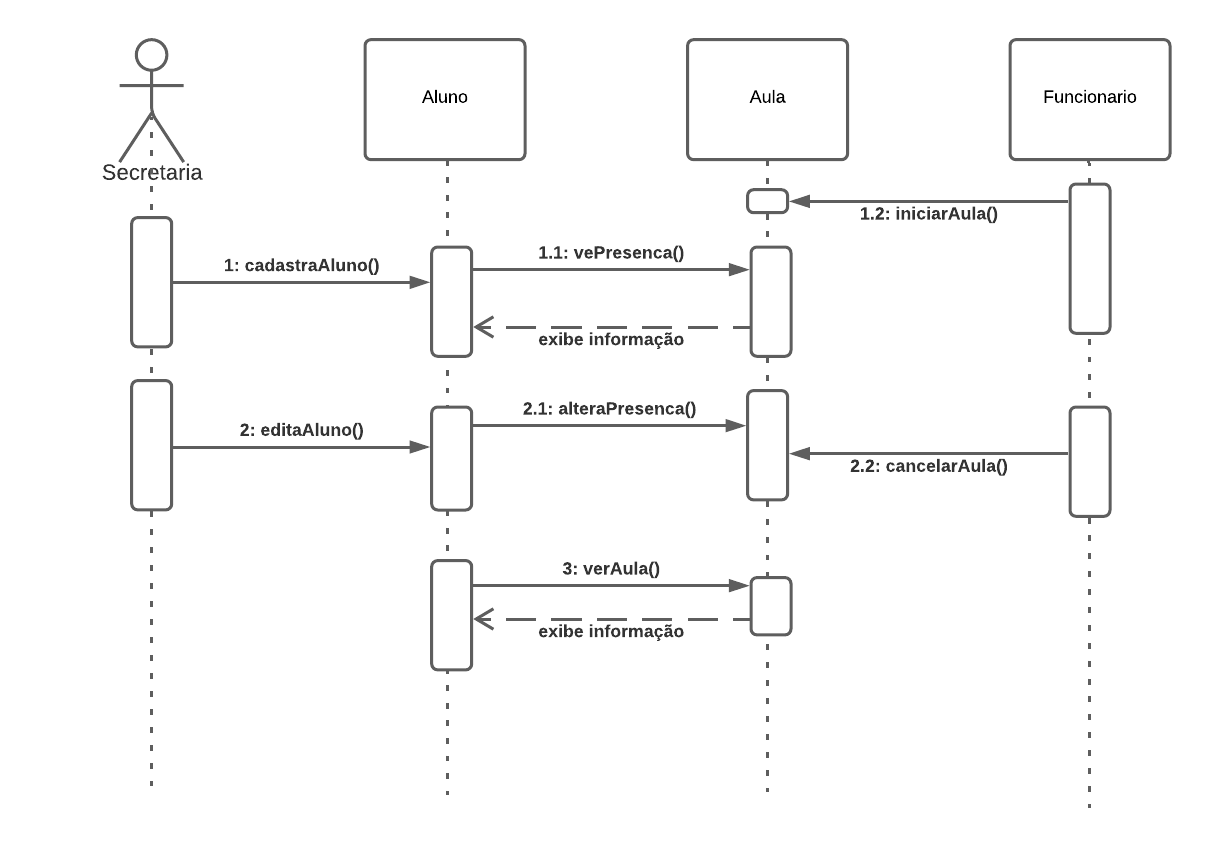
Uma imagem contendo texto

Descrição gerada automaticamente

Fonte: Autores, 2020.

### 2.3.3 Diagrama de sequência

Figura 22 – Diagrama de Sequência



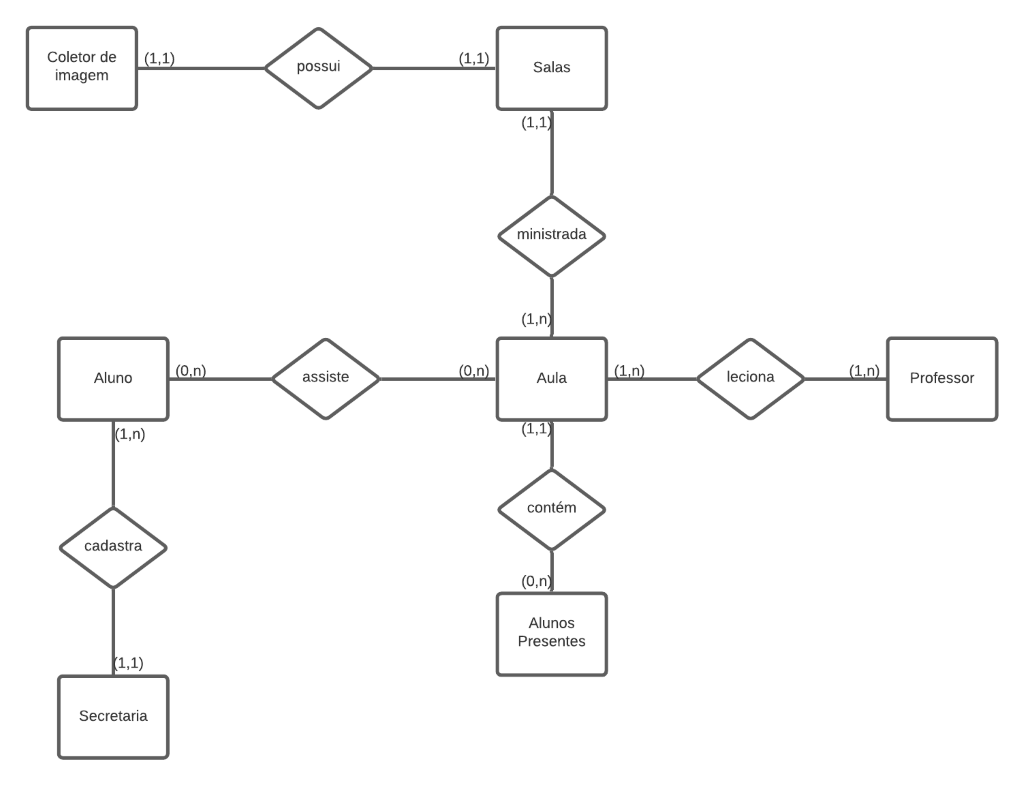
Fonte: Autores, 2020.

## 2.4 Definição de Modelagem de Dados

Modelagem de dados é utilizada para visualizar, por meio de diagramas as definições técnicas da solução, e as estruturas envolvendo a manipulação e o armazenamento de dados. Sendo assim, modelos conceituais de alto nível e modelos físicos podem ser utilizados para auxiliar na identificação das entidades e seus relacionamentos. Para o melhor entendimento da solução proposta neste trabalho, três exemplos de modelagem foram definidos: Diagrama de entidade relacionamento, modelagem relacional e modelagem física.

### 2.4.1 Diagrama de entidade relacionamento (DER):

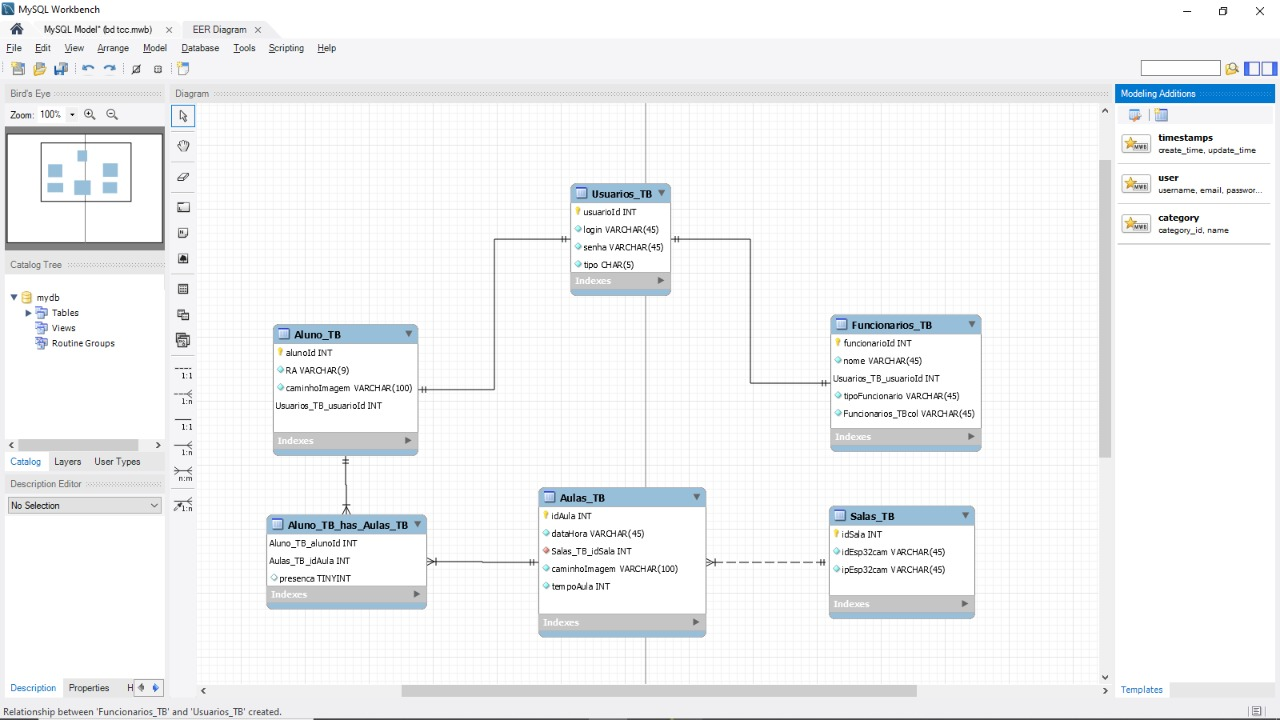
Figura 18 – Diagrama de Entidade Relacionamento



Fonte: Autores, 2020.

### 2.4.2 Modelagem relacional

Figura 19 – Modelagem Relacional



Fonte: Autores, 2020.

### 2.4.3 Modelagem física (SQL)

* Aluno\_TB

**Tabela 11 – Modelagem Física Aluno\_TB**

|  |  |  |
| --- | --- | --- |
| **Nome do campo** | **Tipo do dado** | **Tamanho do campo** |
| alunoId | Numérico | 5 dígitos |
| RA | Alfanumérico | 7 dígitos |
| caminhoImagem | Alfanumérico | 50 dígitos |
| Usuarios\_TB\_usuarioid | Numérico | 5 dígitos |

Fonte: Autores, 2020.

* Usuario\_TB

**Tabela 12 – Modelagem Física Usuario\_TB**

|  |  |  |
| --- | --- | --- |
| **Nome do campo** | **Tipo do dado** | **Tamanho do campo** |
| usuarioId | Numérico | 5 dígitos |
| Login | Alfanumérico | 10 caracteres |
| Senha | Alfanumérico | 10 caracteres |
| Tipo | Alfanumérico | 30 caracteres |

Fonte: Autores, 2020.

* Funcionarios\_TB

**Tabela 13 – Modelagem Física Funcionarios\_TB**

|  |  |  |
| --- | --- | --- |
| **Nome do campo** | **Tipo do dado** | **Tamanho do campo** |
| funcionarioId | Numérico | 5 dígitos |
| Nome | Alfanumérico | 50 caracteres |
| Usuarios\_TB\_usuariosId | Numérico | 5 dígitos |
| tipoFuncionario | Alfanumérico | 30 caracteres |
| Funcionario\_TBcol | Alfanumérico | 50 caracteres |

Fonte: Autores, 2020.

* Aluno\_TB\_has\_Aulas\_TB

**Tabela 14 – Modelagem Física Aluno\_TB\_has\_Aulas\_TB**

|  |  |  |
| --- | --- | --- |
| **Nome do campo** | **Tipo do dado** | **Tamanho do campo** |
| Aluno\_TB\_alunoId | Numérico | 5 dígitos |
| Aulas\_TB\_idAula | Numérico | 5 dígitos |
| Presença | Alfanumérico | 50 caracteres |

Fonte: Autores, 2020.

* Aulas\_TB

**Tabela 15 – Modelagem Física Aulas\_TB**

|  |  |  |
| --- | --- | --- |
| **Nome do campo** | **Tipo do dado** | **Tamanho do campo** |
| idAula | Numérico | 5 dígitos |
| dataHora | Date | 10 caracteres |
| Salas\_TB\_idSala | Numérico | 5 dígitos |
| caminhoImagem | Alfanumérico | 50 caracteres |
| tempoAula | Time | 10 caracteres |

Fonte: Autores, 2020.

* Salas\_TB

**Tabela 16 – Modelagem Física Salas\_TB**

|  |  |  |
| --- | --- | --- |
| **Nome do campo** | **Tipo do dado** | **Tamanho do campo** |
| idSala | Numérico | 5 dígitos |
| idEsp32Cam | Numérico | 5 dígitos |
| ipEsp32Cam | Numérico | 5 dígitos |

Fonte: Autores, 2020.

### 2.4.4 Dicionário de dados:

**Tabela 17 – Dicionario de Dados Aluno\_TB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabela: Aluno\_TB** | | | |
| Campo | Tipo de dados | Tamanho | Descrição |
| alunoId | INT | 2.147.483.647 | Código identificador dos alunos. |
| RA | Varchar(9) | 9 | Código de identificação do aluno na instituição. |
| caminhoImagem | Varchar(100) | 100 | Caminho onde irá esta guardado a imagem do rosto do aluno. |
| Usuarios\_TB\_usuarioId | INT | 2.147.483.647 | Código identificador do usuário |
| Chave primária | alunoId | | |
| Chave estrangeira | Usuarios\_TB\_usuarioId | | |

Fonte: Autores, 2020.

**Tabela 18 – Dicionario de Dados Salas\_TB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabela: Salas\_TB** | | | |
| Campo | Tipo de dados | Tamanho | Descrição |
| idSala | INT | 2.147.483.647 | Código identificador das salas. |
| idEsp32cam | Varchar(45) | 45 | Código de identificação do ESP32 cam instalado na sala |
| ipEsp32cam | Varchar(45) | 45 | IP do ESP32 cam conectado à rede. |
| Chave primária | idSala | | |
| Chave estrangeira | Não contém | | |

Fonte: Autores, 2020.

**Tabela 19 – Dicionario de Dados Aluno\_TB\_has\_Aulas\_TB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabela: Aluno\_TB\_has\_Aulas\_TB** | | | |
| Campo | Tipo de dados | Tamanho | Descrição |
| Aluno\_TB\_alunoId | INT | 2.147.483.647 | Código identificador do aluno. |
| Aulas\_TB\_idAula | int | 2.147.483.647 | Código identificador da aula. |
| presenca | BIT | 1 | Campo marcará a presença do aluno na aula. |
| Chave primária | Aluno\_TB\_alunoId, Aulas\_TB\_idAula | | |
| Chave estrangeira | Não contém | | |

Fonte: Autores, 2020.

**Tabela 20 – Dicionario de Dados Usuarios\_TB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabela: Usuarios\_TB** | | | |
| Campo | Tipo de dados | Tamanho | Descrição |
| usuarioId | INT | 2.147.483.647 | Código identificador do usuario. |
| login | Varchar(45) | 45 | Nome do usuário para logar no sistema. |
| senha | Varchar(45) | 45 | Senha do usuario para logar no sisema. |
| tipo | Char(5) | 5 |  |
| Chave primária | Aluno\_TB\_alunoId | | |
| Chave estrangeira | Não contém | | |

Fonte: Autores, 2020.

**Tabela 21 – Dicionario de Dados Funcionários\_TB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabela: Funcionarios\_TB** | | | |
| Campo | Tipo de dados | Tamanho | Descrição |
| funcionarioId | INT | 2.147.483.647 | Código identificador do funcionario. |
| nome | Varchar(45) | 45 | Nome do funcionario. |
| Usuarios\_TB\_usuarioId | INT | 2.147.483.647 | Código |
| tipoFuncionario | Varchar(45) | 45 | Tipo do funcionário. |
| Chave primária | funcionarioId | | |
| Chave estrangeira | Usuarios\_TB\_usuarioId | | |

Fonte: Autores, 2020.

**Tabela 22 – Dicionario de Dados Aulas\_TB**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabela: Aulas\_TB** | | | |
| Campo | Tipo de dados | Tamanho | Descrição |
| idAula | INT | 2.147.483.647 | Código identificador do aluno. |
| dataHora | VARCHAR(45) | 45 | Código identificador da aula. |
| Salas\_TB\_idSala | INT | 2.147.483.647 | Campo marcará a presença do aluno na aula. |
| caminhoImagem | VARCHAR(100) | 100 |  |
| Chave primária | idAula | | |
| Chave estrangeira | Salas\_TB\_idSala | | |

Fonte: Autores, 2020.

## 2.5 Projeto de Interface Homem Máquina

Para o gerenciamento da solução, foi desenvolvido uma aplicação WEB com interface gráfica, facilitando o uso e controle do sistema. Abaixo está o manual de uso da aplicação.

Figura 11 – Tela de Login



Fonte: Autores, 2020.

Figura 12 – Pagina inicial – visão administrador



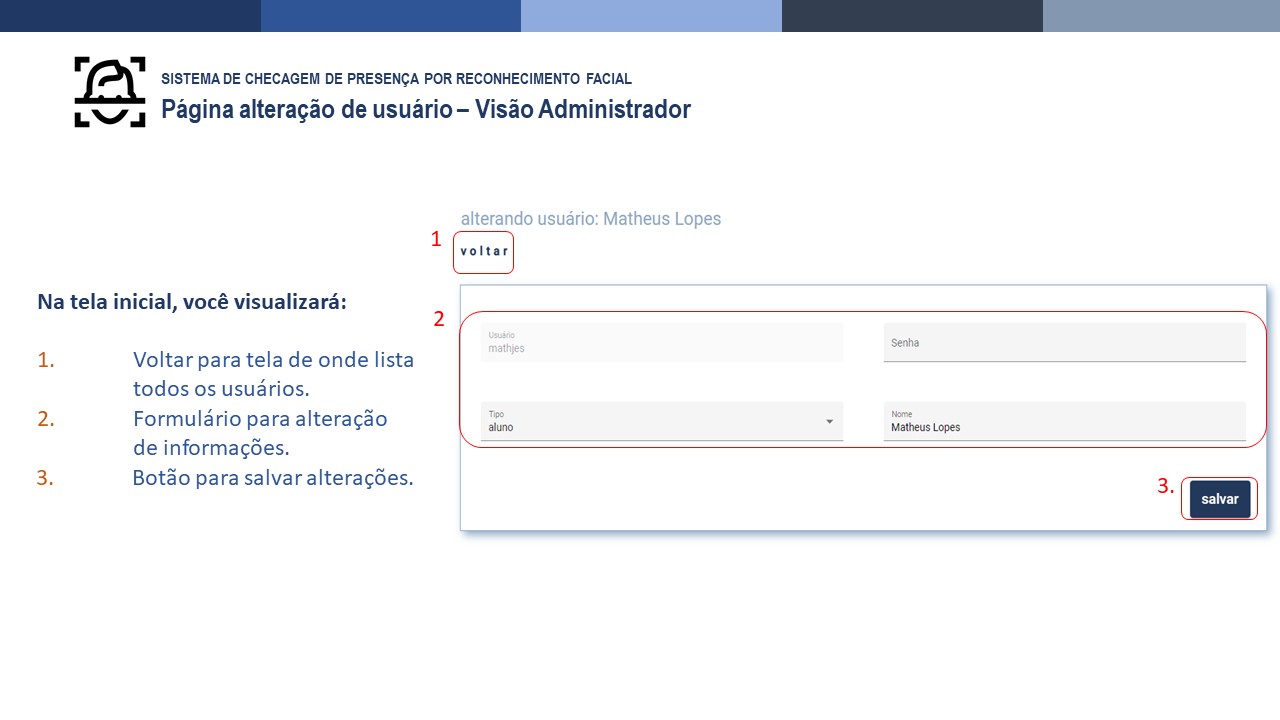
Fonte: Autores, 2020

Figura 13 – Página para edição de usuário .



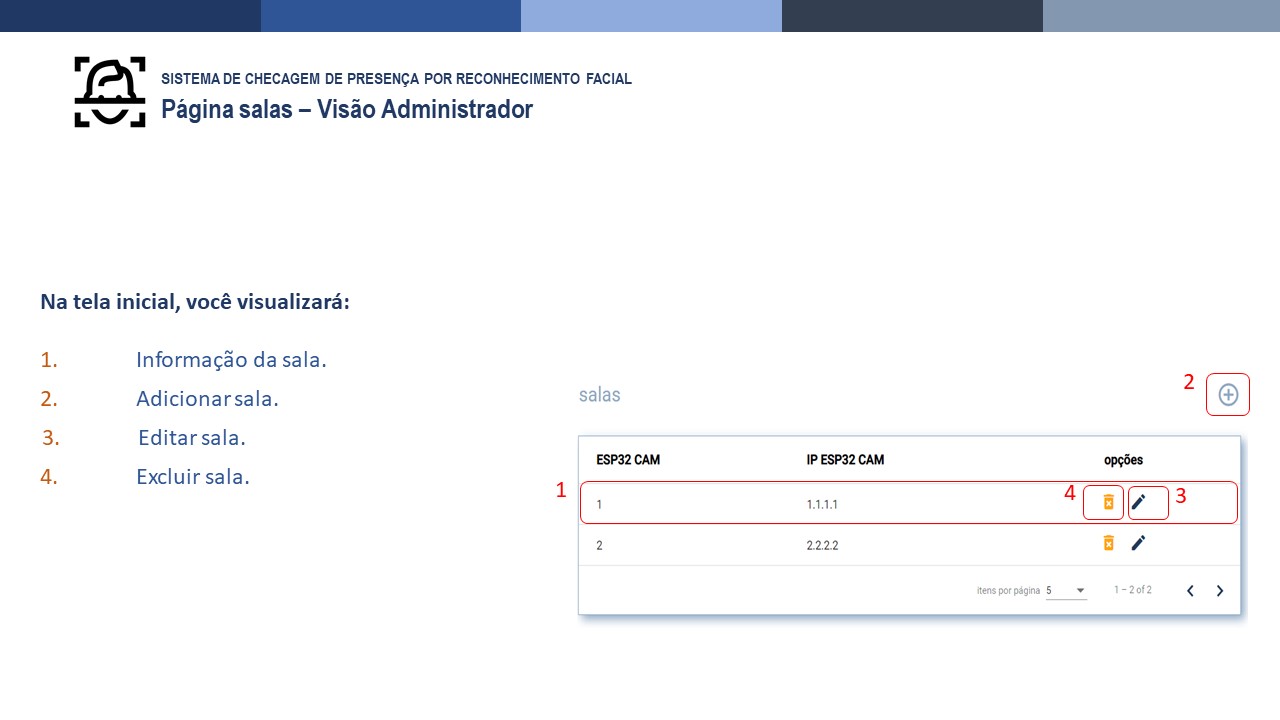
Fonte: Autores, 2020.

Figura 14 – Página para alteração dos dados de um usuário.



Fonte: Autores, 2020.

Figura 15 – Página com a visualização de todas as salas.

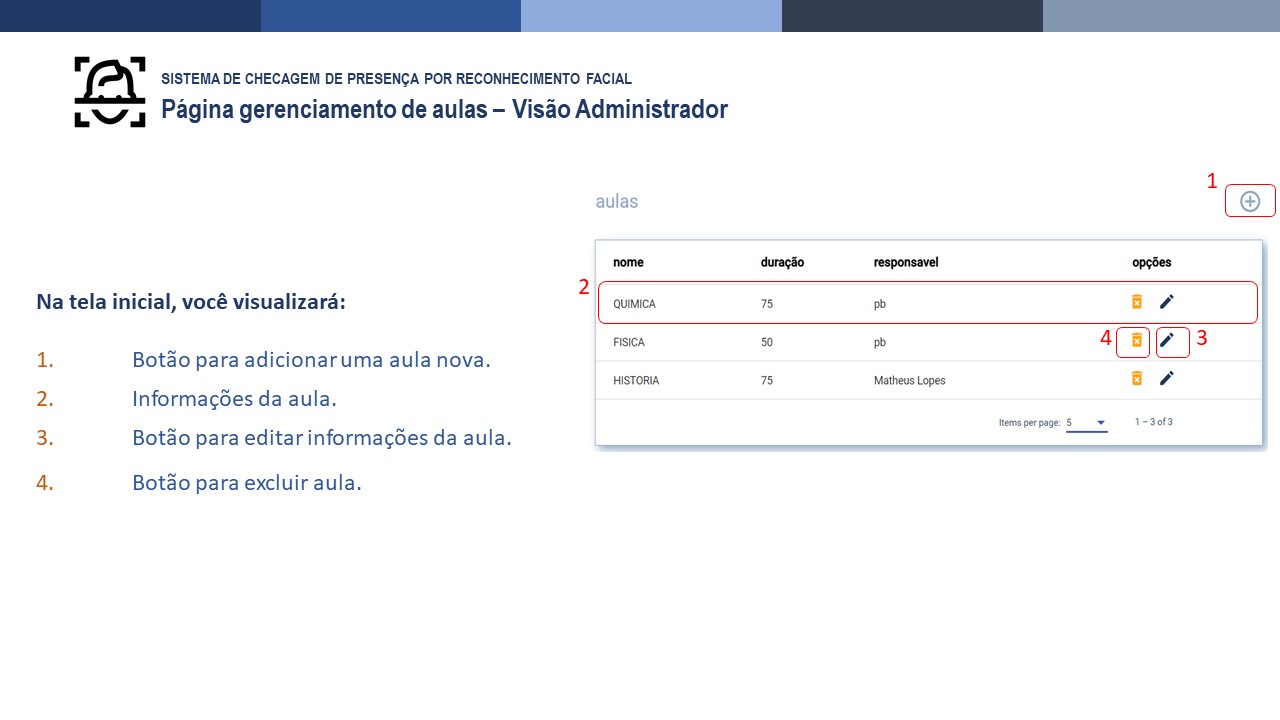


Fonte: Autores, 2020.

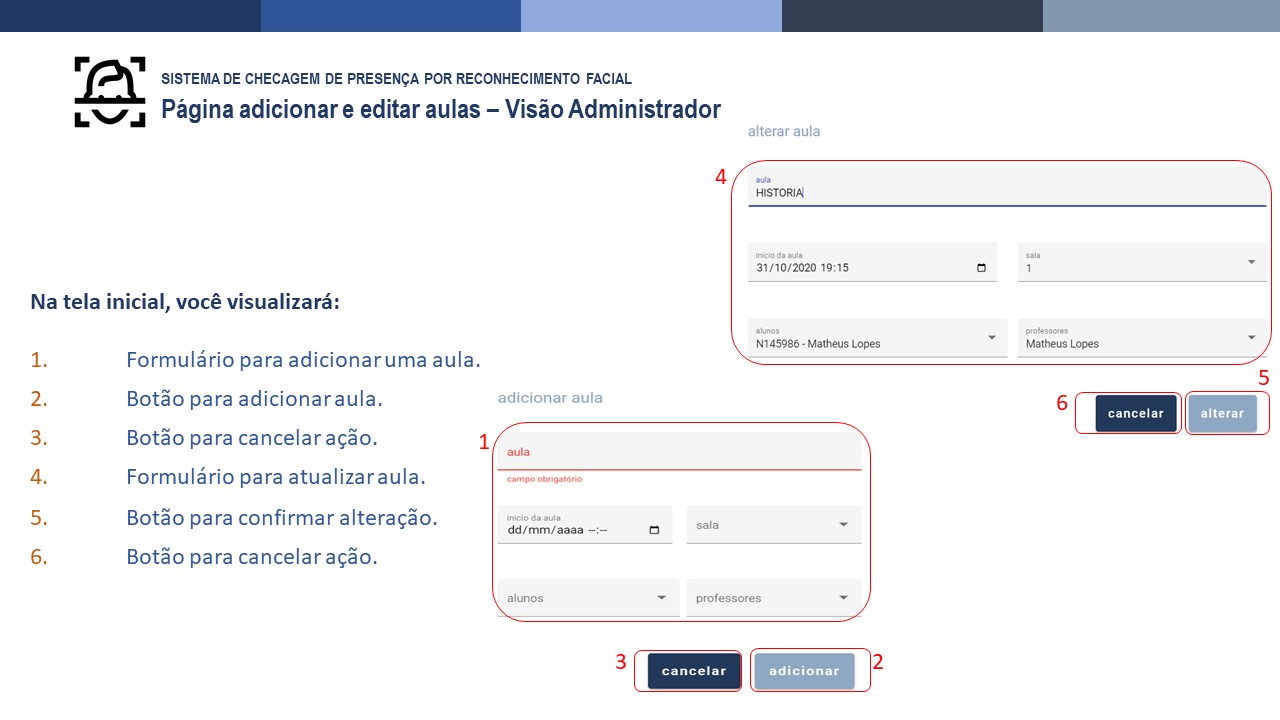
Figura 16 – Página de adição e edição de salas.



Fonte: Autores, 2020.

Figura 17 – Página para o gerenciamento de aula

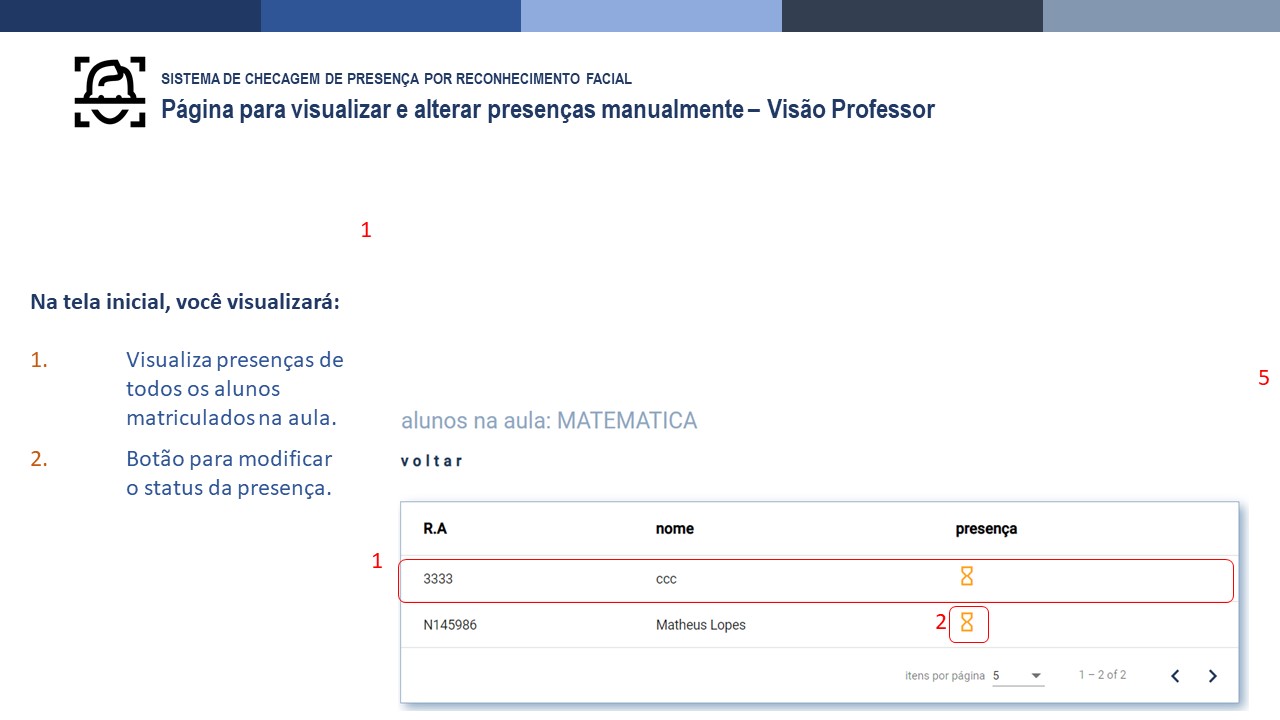
Fonte: Autores, 2020.

Figura 18 – Página para adicionar e editar aulas.

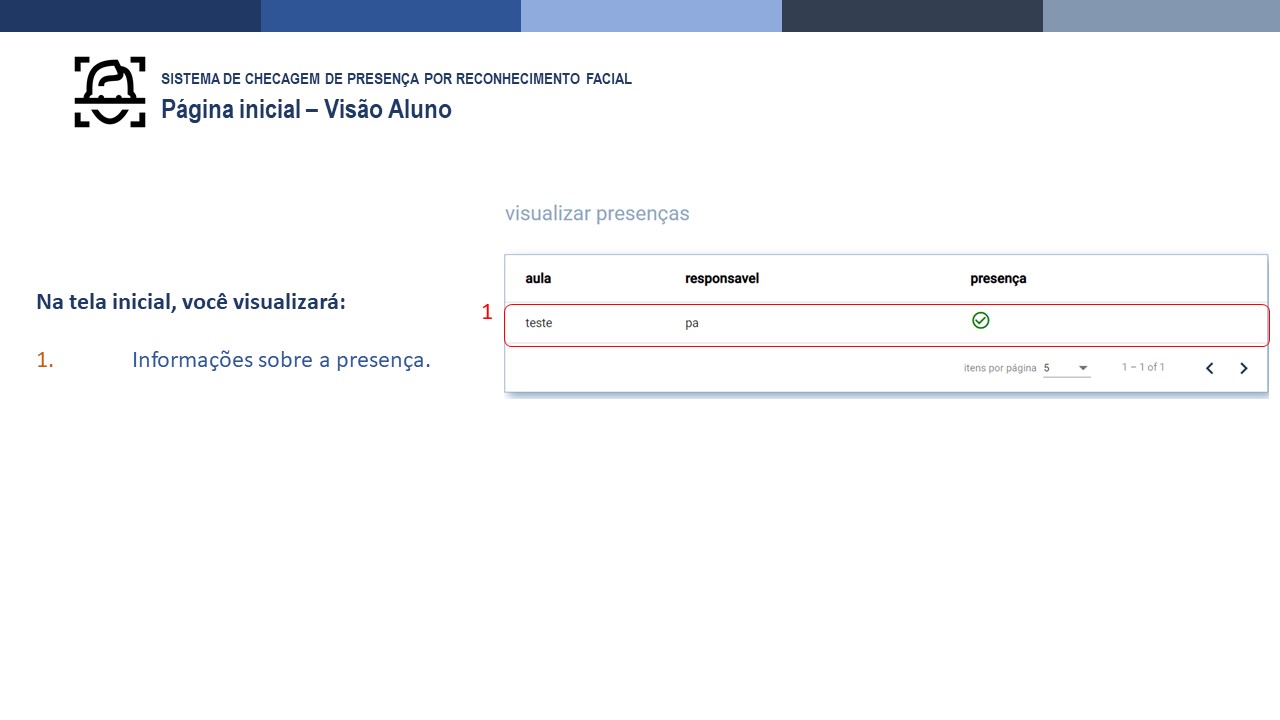
Fonte: Autores, 2020.

Figura 18 – Pagina para visualização de todas as aulas do professor.

Fonte: Autores, 2020.

Figura 18 – Página para visualizar e alterar presenças manualmente.

Fonte: Autores, 2020.

Figura 18 – Página para o aluno visualizar suas presenças.

Fonte: Autores, 2020.

# 3 MÉTODOS E MATERIAIS

Para o desenvolvimento do projeto definido neste trabalho, utilizamos diversas ferramentas e tecnologias que possibilitaram e auxiliaram na qualidade e viabilidade do protótipo.

**3.1 Ferramentas de Software**

Na escolha das ferramentas e tecnologias de software que utilizamos, priorizamos a viabilidade financeira do projeto, sendo assim, focamos em tecnologias gratuitas e/ou de código aberto.

Para o desenvolvimento do código, utilizamos dois ambientes de desenvolvimento integrado, da sigla IDE (*Integrated Development Environment*), o Visual Studio Code para o desenvolvimento do front-end, back-end, e do processamento do reconhecimento facial, e o Arduino IDE, para o desenvolvimento do código embarcado no módulo ESP32-CAM.

Além disso, utilizamos para o desenvolvimento do front-end, ou seja, a camada WEB visível aos usuários do sistema (funcionários da instituição de ensino e alunos), o framework JavaScript Angular, sendo um grande facilitador dessa etapa, além de ser uma ferramenta bastante atual e muito utilizada no mercado. Enquanto isso, para o desenvolvimento do back-end, ou seja, a camada do servidor, que realiza todo o processamento dos dados e executa o reconhecimento facial, utilizamos o ambiente de execução para JavaScript Node.js, o que nos possibilitou trabalhar com uma mesma linguagem em todas as pontas da nossa aplicação WEB, agilizando o desenvolvimento, e também sendo dessa forma um ambiente de execução leve, contribuindo pra versatilidade e viabilidade da implementação da solução em um servidor real, mesmo que com menor capacidade computacional.

Para o armazenamento dos dados do sistema, utilizamos uma instância gratuita do servidor MySQL, um banco de dados relacional que em nossa arquitetura ficou responsável pelo armazenamento dos dados dos usuários e suas contas para acesso, além do registro das aulas, salas, e presenças. A exceção é o armazenamento das imagens, que tem seus arquivos brutos sendo armazenados diretamente no disco de instalação do servidor back-end.

Com o objetivo de reforçar a viabilidade técnica, financeira, e de implementação do sistema em um caso real, optamos pela escolha de conteinerizar a nossa solução, transformando o sistema em uma imagem que pode ser utilizada em qualquer computador que seja escolhido para ser o servidor da instituição de ensino, desde que sua capacidade computacional seja equivalente à necessidade do processamento e do armazenamento para o correto funcionamento do sistema de acordo com a quantidade de alunos e salas da instituição. Para realizar isso, escolhemos a tecnologia gratuita chamada Docker, ferramenta de conteinerização atual e muito utilizada no mercado.

**3.2 Ferramentas de Hardware**

Além das tecnologias digitais que utilizamos e foram descritas no tópico acima, utilizamos ferramentas físicas, chamadas de hardware.

Para o módulo do protótipo, utilizamos o ESP32-CAM, computador de placa única, com Wi-Fi e câmera embutidos, e capacidade de processamento suficiente para coletar uma foto com sua câmera, e devolvê-la através da rede utilizando protocolo HTTP. Visto que o ESP32-CAM não possuí conexão USB, para a inserção do código embarcado no módulo foi necessário o uso de uma placa FTDI FT232RL, responsável pela conversão USB Serial, dessa forma, através de uma simples conexão USB, pudemos inserir o código embarcado para execução no ESP32-CAM.

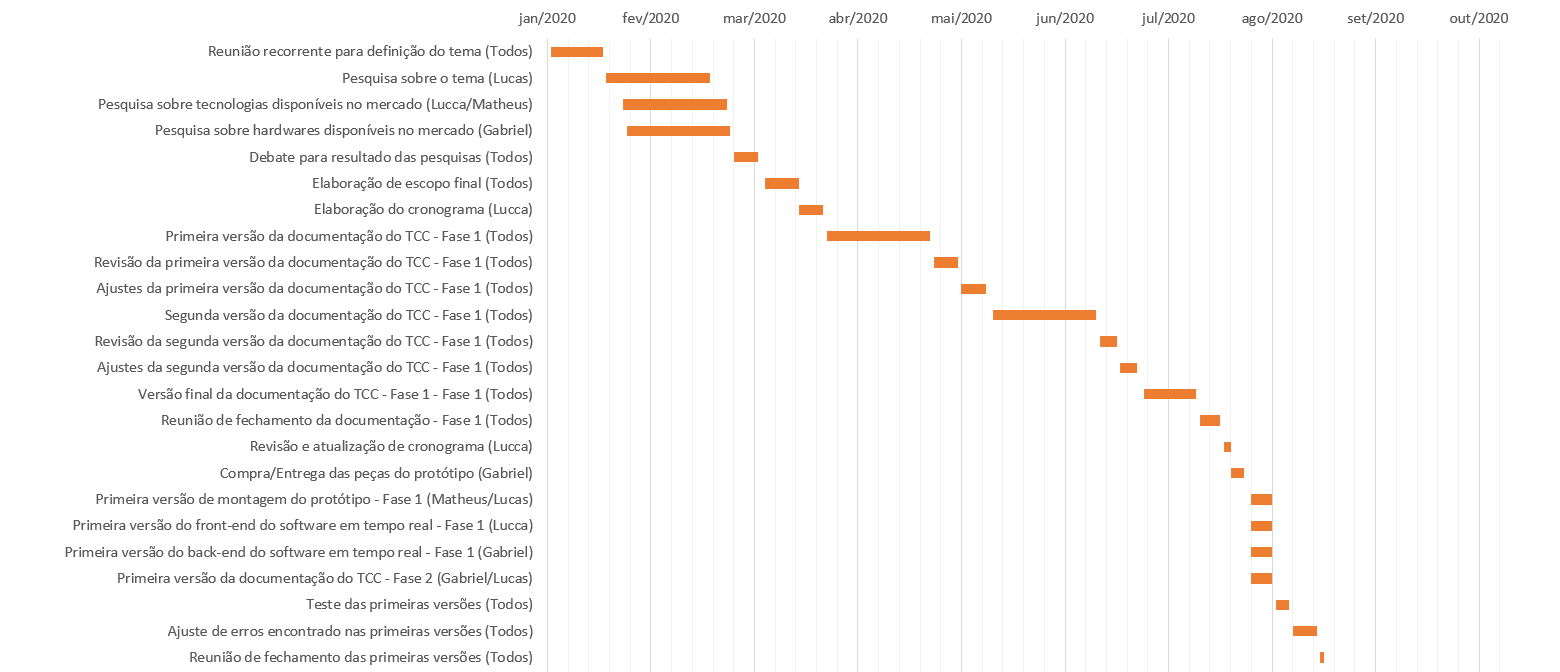
No desenvolvimento e execução do sistema, nós, membros do grupo, utilizamos nossos computadores pessoais, diminuindo assim os custos para a execução do projeto. Para fins de análise da capacidade computacional, abaixo estão as configurações de dois dos computadores utilizados no desenvolvimento.

1. Notebook.
   * Processador Intel i5 6200U (2.3 GHz – 2 núcleos e 4 Threads);
   * 8 GB de Memória RAM;
   * 1 TB de armazenamento (HDD – Disco Rígido);
   * 4 GB de Memória RAM dedicada para processamento gráfico (VRAM).
2. Computador de mesa.
   * Processador AMD A10 5800K (3.8 GHz – 4 núcleos e 4 Threads);
   * 8 GB de Memória RAM;
   * 500 GB de armazenamento (HDD – Disco Rígido);
   * 2 GB de Memória RAM dedicada para processamento gráfico (VRAM).

# 4 RESULTADOS

## 5 CRONOGRAMA

Figura 23 – Cronograma Parte 1



Fonte: Autores, 2020.

Figura 25 – Cronograma Parte 2



Fonte: Autores, 2020.

# 5 CONCLUSÃO

Melhorar a produtividade do antiquado e consolidado sistema de ensino é uma tarefa difícil e que exige atenção à detalhes. É preciso mais do que investir em bons profissionais e boas estruturas, também é necessário se modenizar, se adaptar, e questionar as atuais práticas consideradas comuns nesse sistema. É importante também, inovar o modelo de ensino que se mostra tão antigo e estático, aplicando diversas tecnologias existentes e acessíveis nos dias atuais. Para causar esse impacto no sistema de ensino, se faz necessário extrair o máximo de tempo útil possível de tarefas burocráticas, que não contribuem para o aprendizado de forma direta, e até mesmo de tarefas rotineiras, como a checagem de presença, é possível se extrair tempo útil, com o intuito de melhorar a qualidade geral do modelo, focando os esforços em atividades de ensino prático.

A aplicação da solução proposta neste trabalho indica um impacto positivo em potencial para a sociedade como um todo, e em especial no meio acadêmico, permitindo que mais tempo seja utilizado para atividades benéficas ao aprendizado, e assim, alcançando os objetivos propostos, definindo e desenvolvendo uma solução viável para melhorar a produtividade no modelo de ensino atual.

Essa tarefa árdua de impactar positivamente a produtividade do sistema de ensino é complexa e as soluções propostas devem ser constantemente revisitadas e adaptadas para melhor se encaixarem na realidade prática das instituições de ensino, e portanto, sugerimos que para o prosseguimento do trabalho exercido aqui, em um trabalho futuro, seja estudada a implementação de melhorias na câmera de captação de imagens do módulo, com uma câmera capaz de captar um ambiente maior e muito movimentado, realidade de diversas instituições de ensino.

Esperamos que a aplicação de novas tecnologias no sistema de ensino seja cada vez mais comum, e que através disso possamos evoluir cada vez mais a sociedade.

# 6 TRABALHOS FUTUROS

# 7 REFERÊNCIA BIBLIOGRÁFICA

## 7.1 LEVANTAMENTO BIBLIOGRÁFICO

Neste tópico foram descritas as bibliografias que mais contribuíram com nosso projeto.

**Tabela 24 – Levantamento Bibliográfico “Desenvolvimento de um protótipo de software de reconhecimento facial: estudo do sistema presente”**

|  |  |
| --- | --- |
| **Autor(es):** | Luiz Fernando Peramo, Paulo Cesar de Macedo, Marcos Roberto de Moraes. |
| **Título do trabalho:** | Desenvolvimento de um protótipo de software de reconhecimento facial: estudo do sistema presente!!! |
| **Resumo do trabalho:** | Neste projeto, foi desenvolvido um sistema chamado 'PRESENTE". A ideia central é automatizar o procedimento de controle de presença dos alunos, realizada até então por uma confirmação verbal. Uma regra diferencial neste sistema, é que caso o aluno não compareça dentro do limite de atraso estabelecido, um comunicado será enviado aos responsáveis deste aluno. Notificando - os, da não presença. O sistema foi desenvolvido na linguagem Python e, em um de seus maiores testes obteve 85% de precisão no reconhecimento facial dos alunos. Comprovando, a dispensa da tradicional chamada feita pelos professores. |

Fonte: Autores, 2020.

**Tabela 25 – Levantamento Bibliográfico “Sistemas de Reconhecimento Facial”**

|  |  |
| --- | --- |
| **Autor(es):** | Luiz Filipe Zenicola Braga |
| **Título do trabalho:** | Sistemas de Reconhecimento Facial |
| **Resumo do trabalho:** | Neste projeto foi estudado uma forma de reconhecer rostos, em uma imagem, previamente cadastrados em um banco de dados. O sistema desenvolvido, tem como entrada de dados uma imagem e como saída devolve uma resposta ao usuário se, há ou não, rostos na imagem previamente entregue ao sistema. A detecção facial é feita mediante ao uso do algoritmo denominado "Viola-Jones", tendo como base o treinamento de classificadores mais importantes e, posteriormente, juntando com classificadores mais fracos. Já a coleta de características dos rostos, utilizou - se a "Análise Discriminante", reduzindo a dimensão do espaço enquanto preserva informações que diferencie os rostos. |

Fonte: Autores, 2020.

**Tabela 26 – Levantamento Bibliográfico “Estudo sobre métodos de reconhecimento facial em fotografias digitais”**

|  |  |
| --- | --- |
| **Autor(es):** | Ana Elisa Schmidt, Elvis Cordeiro Nogueira |
| **Título do trabalho:** | Estudo sobre métodos de reconhecimento facial em fotografias digitais |
| **Resumo do trabalho:** | Neste projeto, foi estudado uma maneira de evitar fraudes de identidade por meio de técnicas de reconhecimento facial. Os integrantes do projeto, pensaram em três pilares: Detecção da face, extração de características e reconhecimento da face. A biblioteca utilizada para efetuar o reconhecimento facial, foi a "OpenCV", sendo opensource e desenvolvida pela INTEL. O sistema que seria desenvolvido, ficou para a segunda fase do projeto. |

Fonte: Autores, 2020.

## 7.2 Referências Bibliográficas

**Professores perdem 34% do tempo das aulas com burocracia**.Portal de notícias R7. Disponível em[:](http://noticias.r7.com/educacao/noticias/professores-perdem-34-do-tempo-das-aulas-com-burocracia-20110516.html?question=0) http://noticias.r7.com/educacao/noticias/professores[-perdem-34-do-tempodas-aulas-com-burocracia-](http://noticias.r7.com/educacao/noticias/professores-perdem-34-do-tempo-das-aulas-com-burocracia-20110516.html?question=0)20110516.html?question=0[:](http://noticias.r7.com/educacao/noticias/professores-perdem-34-do-tempo-das-aulas-com-burocracia-20110516.html?question=0). Acesso em: 16 de maio 2011.

**Bringing Facial Recording System To Light**. Organização de pesquisa e discussão sobre inteligência artificial. Disponível em: https://www.partnershiponai.org/facial[-](https://www.partnershiponai.org/facial-recognition-systems/)recognition[-systems/.](https://www.partnershiponai.org/facial-recognition-systems/) Acesso em: 17 de maio de 2020.

**O Papel da Tecnologia**. Portal de notícias Correio Braziliense. Disponível em: https://www.correiobraziliense.com.br/escolhaaescola/papel[-da-tecnologia-escolha-aescola/.](https://www.correiobraziliense.com.br/escolhaaescola/papel-da-tecnologia-escolha-a-escola/) Acesso em: 06 de junho de 2020.

CORDEIRO, Eelvis. **Estudo Sobre Métodos de Reconhecimento Facial em Fotografias Digitais**. 2015. 05 folhas. Bacharel em Sistemas de Informação - FAPESP, Camburiu, 2015.

BRAGA, Luiz. **Sistemas de Reconhecimento Facial**. 2013. 84 folhas. Bacharel em Engenharia da Computação - Universidade de São Carlos, São Carlos, 2013.

**Brasil cai em ranking mundial de educação em matemática e ciências; e fica estagnado em leitura**. Portal de notícias G1. Disponível em: https://g1.globo.com/educacao/noticia/2019/12/03/brasil-cai-em-ranking-mundial-de-educacao-em-matematica-e-ciencias-e-fica-estagnado-em-leitura.ghtml. Acesso em: 19 de setembro de 2020.

**Package Software into Standardized Units for Development, Shipment and Deployment**. Marca detentora e desenvolvedora da tecnologia de conteinerização Docker. Disponível em: https://www.docker.com/resources/what-container. Acesso em: 19 de setembro de 2020.

BRASIL. **Código civil**. Lei nº 9.394 Título 4, Art. 13, Tópico V, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional.

Núñez Novo, Benigno. **O direito de imagem**. Revista JUS Navigandi. Disponível em: https://jus.com.br/artigos/75081/o-direito-de-imagem. Acesso em: 20 de setembro de 2020.

**ESP32: A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications**. Marca ESPRESSIF, detentora e produtora da tecnologia de microcontrolador utilizada (ESP32-CAM). Disponível em: https://www.espressif.com/en/products/socs/esp32. Acesso em: 22 de setembro de 2020

# 8 ANEXOS

|  |  |
| --- | --- |
| **Código-fonte do arquivo:** | ESP32.ino |
| **#include "WiFi.h"**  **#include "esp\_camera.h"**  **#include "esp\_timer.h"**  **#include "img\_converters.h"**  **#include "Arduino.h"**  **#include "soc/soc.h”**  **#include "soc/rtc\_cntl\_reg.h"**  **#include "driver/rtc\_io.h"**  **#include <ESPAsyncWebServer.h>**  **#include <StringArray.h>**  **#include <SPIFFS.h>**  **#include <FS.h>**  **const char\* ssid = "<identificador da rede>";**  **const char\* password = "<senha da rede>";**  **#define CAMERA\_MODEL\_AI\_THINKER**  **AsyncWebServer server(80);**  **boolean takeNewPhoto = false;**  **#define FILE\_PHOTO "/photo.jpg"**  **#define PWDN\_GPIO\_NUM 32**  **#define RESET\_GPIO\_NUM -1**  **#define XCLK\_GPIO\_NUM 0**  **#define SIOD\_GPIO\_NUM 26**  **#define SIOC\_GPIO\_NUM 27**  **#define Y9\_GPIO\_NUM 35**  **#define Y8\_GPIO\_NUM 34**  **#define Y7\_GPIO\_NUM 39**  **#define Y6\_GPIO\_NUM 36**  **#define Y5\_GPIO\_NUM 21**  **#define Y4\_GPIO\_NUM 19**  **#define Y3\_GPIO\_NUM 18**  **#define Y2\_GPIO\_NUM 5**  **#define VSYNC\_GPIO\_NUM 25**  **#define HREF\_GPIO\_NUM 23**  **#define PCLK\_GPIO\_NUM 22**  **const char index\_html[] PROGMEM = R"rawliteral(**  **<!DOCTYPE HTML><html>**  **<head>**  **<meta name="viewport" content="width=device-width, initial-scale=1">**  **<style>**  **body { text-align:center; }**  **.vert { margin-bottom: 10%; }**  **.hori{ margin-bottom: 0%; }**  **</style>**  **</head>**  **<body>**  **<div id="container">**  **<h2>ESP32-CAM Last Photo</h2>**  **<p>It might take more than 5 seconds to capture a photo.</p>**  **<p>**  **<button onclick="rotatePhoto();">ROTATE</button>**  **<button onclick="capturePhoto()">CAPTURE PHOTO</button>**  **<button onclick="location.reload();">REFRESH PAGE</button>**  **</p>**  **</div>**  **<div><img src="saved-photo" id="photo" width="40%"></div>**  **</body>**  **<script>**  **var deg = 0;**  **function capturePhoto() {**  **var xhr = new XMLHttpRequest();**  **xhr.open('GET', "/capture", true);**  **xhr.send();**  **}**  **function rotatePhoto() {**  **var img = document.getElementById("photo");**  **deg += 90;**  **if(isOdd(deg/90)){ document.getElementById("container").className = "vert"; }**  **else{ document.getElementById("container").className = "hori"; }**  **img.style.transform = "rotate(" + deg + "deg)";**  **}**  **function isOdd(n) { return Math.abs(n % 2) == 1; }**  **</script>**  **</html>)rawliteral";**  **void setup() {**  **Serial.begin(115200);**  **WiFi.begin(ssid, password);**  **while (WiFi.status() != WL\_CONNECTED) {**  **delay(1000);**  **Serial.println("Connecting to WiFi...");**  **}**  **if (!SPIFFS.begin(true)) {**  **Serial.println("An Error has occurred while mounting SPIFFS");**  **ESP.restart();**  **}**  **else {**  **delay(500);**  **Serial.println("SPIFFS mounted successfully");**  **}**  **Serial.print("IP Address: http://");**  **Serial.println(WiFi.localIP());**  **WRITE\_PERI\_REG(RTC\_CNTL\_BROWN\_OUT\_REG, 0);**  **camera\_config\_t config;**  **config.ledc\_channel = LEDC\_CHANNEL\_0;**  **config.ledc\_timer = LEDC\_TIMER\_0;**  **config.pin\_d0 = Y2\_GPIO\_NUM;**  **config.pin\_d1 = Y3\_GPIO\_NUM;**  **config.pin\_d2 = Y4\_GPIO\_NUM;**  **config.pin\_d3 = Y5\_GPIO\_NUM;**  **config.pin\_d4 = Y6\_GPIO\_NUM;**  **config.pin\_d5 = Y7\_GPIO\_NUM;**  **config.pin\_d6 = Y8\_GPIO\_NUM;**  **config.pin\_d7 = Y9\_GPIO\_NUM;**  **config.pin\_xclk = XCLK\_GPIO\_NUM;**  **config.pin\_pclk = PCLK\_GPIO\_NUM;**  **config.pin\_vsync = VSYNC\_GPIO\_NUM;**  **config.pin\_href = HREF\_GPIO\_NUM;**  **config.pin\_sscb\_sda = SIOD\_GPIO\_NUM;**  **config.pin\_sscb\_scl = SIOC\_GPIO\_NUM;**  **config.pin\_pwdn = PWDN\_GPIO\_NUM;**  **config.pin\_reset = RESET\_GPIO\_NUM;**  **config.xclk\_freq\_hz = 20000000;**  **config.pixel\_format = PIXFORMAT\_JPEG;**  **if (psramFound()) {**  **config.frame\_size = FRAMESIZE\_UXGA;**  **config.jpeg\_quality = 10;**  **config.fb\_count = 2;**  **} else {**  **config.frame\_size = FRAMESIZE\_SVGA;**  **config.jpeg\_quality = 12;**  **config.fb\_count = 1;**  **}**  **esp\_err\_t err = esp\_camera\_init(&config);**  **if (err != ESP\_OK) {**  **Serial.printf("Camera init failed with error 0x%x", err);**  **ESP.restart();**  **}**  **server.on("/", HTTP\_GET, [](AsyncWebServerRequest \* request) {**  **request->send\_P(200, "text/html", index\_html);**  **});**  **server.on("/capture", HTTP\_GET, [](AsyncWebServerRequest \* request) {**  **takeNewPhoto = true;**  **request->send\_P(200, "text/plain", "Taking Photo");**  **});**  **server.on("/saved-photo", HTTP\_GET, [](AsyncWebServerRequest \* request) {**  **request->send(SPIFFS, FILE\_PHOTO, "image/jpg", false);**  **});**  **server.begin();**  **}**  **void loop() {**  **if (takeNewPhoto) {**  **capturePhotoSaveSpiffs();**  **takeNewPhoto = false;**  **}**  **delay(1);**  **}**  **bool checkPhoto( fs::FS &fs ) {**  **File f\_pic = fs.open( FILE\_PHOTO );**  **unsigned int pic\_sz = f\_pic.size();**  **return ( pic\_sz > 100 );**  **}**  **void capturePhotoSaveSpiffs( void ) {**  **camera\_fb\_t \* fb = NULL;**  **bool ok = 0;**  **do {**  **Serial.println("Taking a photo...");**  **fb = esp\_camera\_fb\_get();**  **if (!fb) {**  **Serial.println("Camera capture failed");**  **return;**  **}**  **Serial.printf("Picture file name: %s\n", FILE\_PHOTO);**  **File = SPIFFS.open(FILE\_PHOTO, FILE\_WRITE);**  **if (!file) {**  **Serial.println("Failed to open file in writing mode");**  **}**  **else {**  **file.write(fb->buf, fb->len);**  **Serial.print("The picture has been saved in ");**  **Serial.print(FILE\_PHOTO);**  **Serial.print(" - Size: ");**  **Serial.print(file.size());**  **Serial.println(" bytes");**  **}**    **file.close();**  **esp\_camera\_fb\_return(fb);**  **ok = checkPhoto(SPIFFS);**  **} while ( !ok );**  **}** | |