

**UNIVERSIDADE PAULISTA
CIÊNCIA DA COMPUTAÇÃO**

ALLAN DE SOUSA BRAGA

LETICIA GOMES COSTA

PEDRO HENRIQUE SILVA DIAS

RHUAN ALVES MEIRA SANTOS

**A MELHORIA DE ATENDIMENTO COM BASE NA DETECÇÃO DE
EMOÇÕES VIA RECONHECIMENTO FACIAL PARA A EVOLUÇÃO DAS
EMPRESAS**

**SÃO PAULO
2022**

ALLAN DE SOUSA BRAGA

LETICIA GOMES COSTA

PEDRO HENRIQUE SILVA DIAS

RHUAN ALVES MEIRA SANTOS

**A MELHORIA DE ATENDIMENTO COM BASE NA DETECÇÃO DE
EMOÇÕES VIA RECONHECIMENTO FACIAL PARA A EVOLUÇÃO DAS
EMPRESAS**

**Trabalho de Conclusão de Curso
apresentado ao Curso de Ciência da
Computação da Universidade Paulista, como
requisito parcial para obtenção do grau de
Graduação em Ciência da Computação.**

Orientador: Prof. Me. Marco Antônio Gomes

SÃO PAULO

2022

ALLAN DE SOUSA BRAGA

LETICIA GOMES COSTA

PEDRO HENRIQUE SILVA DIAS

RHUAN ALVES MEIRA SANTOS

**A MELHORIA DE ATENDIMENTO COM BASE NA DETECÇÃO DE
EMOÇÕES VIA RECONHECIMENTO FACIAL PARA A EVOLUÇÃO DAS
EMPRESAS**

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Paulista.

Orientador: Prof. Me. Marco Antônio Gomes

Aprovado em:

BANCA AVALIADORA

Prof. Me. André Gomes de Lira Muniz

Prof. Me. Jose Luiz Valente Junior

Prof. Me. Paulo Rogerio da Silva

RESUMO

No atual mercado, é perceptível a crescente necessidade por parte das pequenas, médias e grandes empresas, do aperfeiçoamento contínuo no serviço de atendimento ao consumidor, tendo isso em vista, é de suma importância que seja realizada uma retenção e ampla análise de todos os dados gerados pelos atendimentos prestados aos clientes, dessa maneira, buscando obter meios de inicialização à otimização de serviços prestados à consumidores em instituições públicas e privadas que possuam atendimentos presenciais.

Dado o exposto, com o auxílio de Inteligência Artificial integrada à um componente físico de Internet Das Coisas (IoT), temos como proposta realizar a captura de imagens e interpretação de micro expressões faciais expostas pelo usuário no decorrer de seu atendimento, atingindo o objetivo de ter a capacidade de análise de microexpressões faciais para determinação de emoções do consumidor em relação à seu atendimento com maior acurácia, dados estes, que podem ser utilizados para fins de qualidade.

Palavras-Chaves: Emoções, Microexpressões, Inteligência Artificial, IOT, Reconhecimento Emocional.

ABSTRACT

In the current market, the growing need on the part of small, medium and large companies, for continuous improvement in customer service is noticeable, with this in mind, it is of paramount importance that a retention and extensive analysis of all data be carried out. generated by services provided to customers, in this way, seeking to obtain means of initializing the optimization of services provided to consumers in public and private institutions that had face-to-face services.

Given the above, with the help of Artificial Intelligence integrated into a physical component of the Internet of Things (IoT), we propose to capture images and interpret micro facial expressions exposed by the user during their service, achieving the objective of have the ability to analyze facial microexpressions to determine consumer emotions in relation to their service with greater accuracy, data that can be used for quality purposes.

Keywords: Emotions, Microexpressions, Artificial Intelligence, IoT, Emotional Recognition.

LISTA DE ABREVIATURAS E SIGLAS

FACS *Facial Action Coding System;*

IA *Inteligência Artificial;*

ADA *American Psychological Association;*

EMFACS *Emotion Facial Action Coding System;*

AU *Action Units;*

UA *Unidades de Ação;*

RNA *Redes Neurais Artificiais;*

CNN *Convolutional Neural Network;*

RNC *Rede Neural Convolucional;*

CPU *Central Processing Unit;*

GPU *Graphics Processing Units;*

ASIC *Application-Specific Integrated Circuit;*

TPU *Tensor Processing Units;*

NN *Neural networks;*

IDE *Integrated Development Environment;*

AWS *Amazon Web Services;*

RDS *Relational Database Service;*

RAM *Random Access Memory;*

SGBD *Sistema de Gerenciamento de Banco de Dados;*

3D *Tridimensionalidade;*

API *Application Programming Interface;*

REST *Representational State Transfer;*

LGPD *Lei Geral de Proteção de Dados;*

IoT *Internet of Things;*

UX *User Experience;*

UI *User Interface;*

CONTECSI *International Conference on Information Systems and Technology Management;*

LED *Light Emitting Diode;*

SD *Secure Digital;*

Mb *Megabytes;*

Kb *Kilobytes;*

ROM *Read Only Memory;*

RAM *Random Access Memory;*

SRAM *Static Random Access Memory;*

BIT *Binary Digit;*

PWM *Pulse Width Modulation;*

EEPROM *Electrically Erasable Programmable Read-Only Memory;*

DEVOPS *Desenvolvimento e Operação;*

LISTA DE FIGURAS

Figura 1 - Expressões universais por Paul Ekman e Wallace Friesen	21
Figura 2 - Características que representam a microexpressão de Surpresa ...	22
Figura 3 - Características que representam a microexpressão de Medo	23
Figura 4 - Características que representam a microexpressão de Aversão.....	24
Figura 5 - Características que representam a microexpressão de Raiva	25
Figura 6 - Características que representam a microexpressão de Felicidade..	26
Figura 7 - Características que representam a microexpressão de Tristeza	26
Figura 8 - Características que representam a microexpressão de Desprezo...	27
Figura 9 - Modelo de neurônio	30
Figura 10 - Funções de ativação	30
Figura 11 - Rede Neural Profunda	31
Figura 12 - GoogLeNet.....	32
Figura 13 - Top 5 classificações ImageNet	33
Figura 14 - Filtros de Camadas Convolucionais.....	34
Figura 15 - Filtros na Detecção de Objetos.....	35
Figura 16 - Representação Docker	41
Figura 17 - Arquitetura monolítica e microserviços.....	44
Figura 18 - Imagem ESP32 CAM.....	48
Figura 19 - Arquitetura de Software N camadas em API Bridge	50
Figura 20 - Comandos REST e suas ações	52
Figura 21 - Rota para obter classificações de atendimentos de colaborador ...	53
Figura 22 - Rota para obter classificações de atendimentos de organização ..	54

Figura 23 - Rota para enviar imagens de atendimentos.....	55
Figura 24- Arquitetura de Software camada única em API IA	56
Figura 25 - Rota para análise em API IA.....	57
Figura 26 - Modelagem do Bando de Dados.....	58
Figura 27 - Diagrama de integração de componentes do projeto.....	59
Figura 28 - QR Code para acesso ao repositório	62
Figura 29 - Desempenho e Classificação do Aprendizado de Máquina	63
Figura 30 - Definição de <i>callbacks</i> e montagem de logs	63
Figura 31 - Treinamento modelo de rede neural	64
Figura 32 - Face triste	64
Figura 33 - Face assustada.....	65
Figura 34 - Face feliz.....	65

LISTA DE TABELAS

Tabela 1 - Requisitos funcionais 45

Tabela 2 - Requisitos não-funcionais 46

SUMÁRIO

1 INTRODUÇÃO	12
1.1 MOTIVAÇÃO.....	12
1.2 OBJETIVOS.....	13
1.2.1 OBJETIVO GERAL.....	13
1.2.2 OBJETIVO ESPECÍFICO	13
 2 PROCEDIMENTOS METODOLÓGICOS.....	15
2.1 CARACTERIZAÇÃO DA PESQUISA	15
2.2 PROPOSTA DA SOLUÇÃO.....	15
2.3 DELIMITAÇÕES.....	15
2.4 TRABALHOS RELACIONADOS	15
 3 FUNDAMENTAÇÃO TEÓRICA	17
3.1 O ESTUDO DAS EMOÇÕES.....	17
3.2 EXPRESSÕES FACIAIS.....	18
3.3 FACS E MICROEXPRESSÕES	19
3.3.1 FACS	19
3.3.2 MICROEXPRESSÕES	20
3.3.2.1 Surpresa.....	21
3.3.2.2 Medo	22
3.3.2.3 Aversão	23
3.3.2.4 Raiva	24
3.3.2.5 Felicidade	25
3.3.2.6 Tristeza.....	26
3.3.2.7 Desprezo	27

3.4 INTELIGÊNCIA ARTIFICIAL	27
3.4.1 MACHINE LEARNING.....	28
3.4.2 REDE NEURAL ARTIFICIAL	28
3.4.2.1 Neurônio Artificial	29
3.4.3 REDES NEURAIS CONVOLUCIONAIS.....	31
3.4.3.1 GOOGLNET	32
3.4.3.2 ALEXNET	32
3.4.3.3 Camadas	33
3.4.3.3.1 Camada de Convulação.....	34
3.4.3.3.2 Camada de Pooling.....	34
3.4.3.3.3 Camada Totalmente Conectada.....	34
3.4.3.3.4 Percepção Convolucional.....	34
3.4.3.4 Função de Ativação.....	35
3.4.3.4.1 Função Linear	35
3.4.3.4.2 Função Sigmóide.....	36
3.4.3.4.3 Função Tanh	36
3.4.3.4.4 Função ReLU	36
3.5 OPENCV	36
3.6 TENSORFLOW.....	36
3.7 KERAS.....	37
 4 IMPLEMENTAÇÃO	 38
4.1 MATERIAIS E MÉTODOS	38
4.1.1 FERRAMENTAS UTILIZADAS.....	38
4.1.1.1 Arduino IDE	38
4.1.1.2 Microsoft Visual Studio	38
4.1.1.3 C++.....	39
4.1.1.3 C# (CSharp)	39

4.1.1.4 .NET Core	40
4.1.1.5 Docker	40
4.1.1.6 Heroku	41
4.1.1.7 MySql	41
4.1.1.8 AWS RDS.....	42
4.1.1.9 Google Colab	42
4.1.1.10 Jupyter notebooks	42
4.1.1.11 Kisslog	42
4.1.1.12 Swagger	43
4.1.2 ARQUITETURA DE SOLUÇÃO UTILIZADA	44
4.1.2.1 Arquitetura de Microserviços	44
4.1.2 REQUISITOS DE SOFTWARE	44
4.1.2.1 Requisitos Funcionais	45
4.1.2.2 Requisitos Não-Funcionais.....	45
4.2 DESENVOLVIMENTO.....	47
4.2.1 INTELIGÊNCIA ARTIFICIAL.....	47
4.2.1.1 <i>Dataset</i>	47
4.2.1.2 Modelo.....	47
4.2.2 HARDWARE.....	47
4.2.2.1 ESP32 Cam.....	48
4.2.3 SOFTWARE	49
4.2.3.1 API Bridge	49
4.2.3.1.1 <i>Arquitetura do software</i>	49
4.2.3.1.2 <i>REST</i>	51
4.2.3.2 API IA	55
4.2.3.2.1 <i>Arquitetura do software</i>	56

4.2.3.2.1 REST.....	56
4.3.4 BANCO DE DADOS	58
4.2.4 DIAGRAMA DE INTEGRAÇÃO	59
4.2.4.1 Usabilidade.....	60
4.2.4.2 Proteção de dados	61
4.2.5 REPOSITÓRIOS	62
4.3 RESULTADOS	62
4.4 TRABALHOS FUTUROS.....	66
5 CONSIDERAÇÕES FINAIS	67
REFERÊNCIAS BIBLIOGRÁFICAS.....	68

1 INTRODUÇÃO

A qualidade de atendimento das empresas perante seus consumidores é uma pauta importante a ser discutida. Um bom atendimento é mais importante para consumidor do que qualidade, revela pesquisa. Revista Pequenas Empresas & Grandes Negócios, 08 dez. 2009. Com base nisso, identificamos que a qualificação profissional de colaboradores é o caminho de sucesso para as empresas, pois a fidelização do cliente se torna uma consequência, a partir do momento em que sua experiência consiste em um bom atendimento, afinal, o contágio emocional, fenômeno que ocorre quando as emoções e o comportamento de uma pessoa afetam as emoções e o comportamento de outra, é uma manifestação multiplicadora que envolve atributos psicológicos, fisiológicos, comportamentais e social de forma inconsciente e automática. Dessa maneira, desencadeando a sincronia emocional, de atenção e comportamento (HATFIELD, CACIOPPE E RAPSON, 1994).

Através da constante busca de excelência nos processos de atendimento ao público, foi observado que as expressões faciais inconscientes durante o decorrer de um atendimento, são ótimas fontes de feedbacks que os clientes fornecem. Sob esse viés, foi idealizado um sistema para análise de expressões, baseado em reconhecimento facial com propósito de detecção de emoções capaz de definir a experiência do usuário em um atendimento.

1.1 MOTIVAÇÃO

A capacidade de processamento dos computadores nos últimos anos aumentou a um nível que possibilitou criar e aprimorar técnicas para auxiliar na resolução de problemas, uma destas técnicas é a aprendizagem de máquina, Machine Learning, mas outro fator que também colocou a aprendizagem de máquina em evidência foi a facilidade de utilizar de ambientes remotos para processar uma grande quantidade de dados sem precisar ter uma estrutura própria. Esses ambientes remotos facilitaram o acesso de pessoas com algum interesse ou de outras áreas que não a computação. Empresas de menor porte também utilizam esse tipo de ambiente para criar e testar novas soluções,

facilitando o acesso que antes necessitava de máquinas com grande poder de processamento para iniciar em certos assuntos que exigem um pouco mais.

O atendimento ao cliente sempre traz inúmeras maneiras de avaliar os serviços prestados e buscar indicadores de problemas em produtos e serviços, comumente sendo representado pela pesquisa de satisfação no final do atendimento, o que se mostra não sendo muito prático e confiável, trazendo muitas vezes uma burocracia extra indesejada. Com a implementação de um sistema de análise e reconhecimento de emoções existe a possibilidade de evitar alguns destes problemas.

As emoções expressadas em certas situações mudam conforme o contexto social em que o indivíduo está inserido. Porém, algumas expressões faciais podem ser consideradas como sendo universais (EKMAN, 2003). Devido a este aspecto as emoções expressadas são uma fonte bem confiável de dados, podendo até serem utilizadas em conjunto com outras formas de avaliação, dando uma ótima base para criar relatórios para análise de atendimento, produtos e serviços.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Avaliar o desempenho de atendimentos presenciais ao público com a identificação de emoções baseado no reconhecimento facial por meio de Inteligência Artificial, sobre uma arquitetura de rede neural treinada para a identificação de microexpressões da face humanas, visando o aperfeiçoamento do atendimento e satisfação do consumidor com base em sua experiência prévia.

1.2.2 OBJETIVO ESPECÍFICO

Analisar as características que permitem identificar qual modelagem apresenta melhor resultado em relação às emoções abordadas.

Considerando o desenvolvimento do trabalho e o objetivo geral apresentado, destacam-se os seguintes objetivos específicos:

- Utilizar o reconhecimento facial para detecção das emoções expressadas fundamentado nas imagens da base. As emoções presentes na base são: neutro, felicidade, tristeza, raiva, medo, aversão e surpresa;
- Realizar uma melhoria nos atendimentos ao consumidor utilizando microexpressões e Facial Action Coding System (FACS), que permite identificar características importantes nas expressões faciais humanas;
- Compreender qual emoção foi expressa pelo usuário durante seu atendimento;
- Treinar uma inteligência artificial em Python com Colab Notebook para reconhecer emoções;

2 PROCEDIMENTOS METODOLÓGICOS

Metodologia de pesquisa é o estudo de métodos e técnicas, e linha de seguimento a ser percorrida dentro de um estudo. O presente artigo tem como abordagem o método de pesquisa qualitativa, pois tem como base a aplicação de dados não mensuráveis e para fins descritivos.

2.1 CARACTERIZAÇÃO DA PESQUISA

Essa pesquisa qualitativa procura entender o comportamento do consumidor, com base em suas propriedades e experiências individuais.

O estudo foi produzido para conhecimento teórico a respeito da relevância dos sentimentos e emoções expressadas diante de uma situação, mas também para propiciar uma solução de unificação para alinhar esses sentimentos aos resultados de satisfação do público ao atendimento prestado por um estabelecimento.

2.2 PROPOSTA DA SOLUÇÃO

A proposta para o presente artigo é o uso de uma câmera para captar as expressões faciais de um consumidor após a prestação de um atendimento, em distintas iluminações e ângulos onde serão encaminhadas para a IA (Inteligência Artificial) analisar os sentimentos expressos em sua face, detectando suas emoções como fonte de feedback ao atendimento realizado.

2.3 DELIMITAÇÕES

O artigo delimitou-se a análise de identificação facial de emoções com a utilização de uma IA (Inteligência Artificial) e idealização de projeto de hardware e software.

2.4 TRABALHOS RELACIONADOS

Há mais de uma década, a tecnologia de análise de expressões faciais é de interesse das empresas e pesquisadores, em 2012 a rede de supermercados Walmart registrou uma patente referente a reconhecimento de emoções para otimizar a experiência dos clientes durante as compras, analisando se há

insatisfação no caixa e caso ocorra, entrar em contato para analisar os problemas relacionados.

Tendo visto que empresas do porte do Walmart estariam entrando neste mercado, apareceram empresas que fornecem serviços relacionados ao mercado de análise facial, dentre elas a NVISO, que fornece uma solução de analisar em tempo real as reações do usuário enquanto ele usa algum aplicativo através da câmera do dispositivo utilizado para o acesso.

Outra abordagem interessante é o protótipo publicado na revista H-TEC que busca utilizar a análise facial para ajudar familiares e cuidadores de pessoas com mobilidade ou fala reduzida a identificar problemas como desconforto, dor e afins utilizando uma webcam.

Um trabalho publicado na CONTECSI de 2020, tem como objetivo capturar e analisar as expressões faciais dos colaboradores em tempo real durante a jornada de trabalho. para facilitar decisões nas avaliações e embasar possíveis encaminhamentos para profissionais especializados.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 O ESTUDO DAS EMOÇÕES

As emoções são comportamentos que os seres humanos vivenciam em resposta de acontecimentos e contextos identificadas pelo rosto por meio das expressões faciais. Segundo a ADA, American Psychological Association, o tipo de emoção que uma pessoa transmite é determinado pela condição que desencadeia a emoção, por exemplo, uma pessoa se sente contente ao ser bem recebida ou se sente triste ao não ter atenção. O desencadeio de emoções se dá a experiência de ter uma reação emocional em mídia ou ambiente social. As emoções governam diariamente as vidas das pessoas, visto que decisões são tomadas baseadas em emoções momentâneas, estando felizes, tristes, entediados, zangados ou frustrados. Entender as emoções pode ajudar a ter uma vida com maior facilidade e estabilidade, pois hobbies e atividades, por exemplo, são escolhidos com base nas emoções que eles provocam.

Segundo o livro *Discovering Psychology* (1990), a emoção é um estado psicológico complexo que envolve três componentes distintos: em primeiro lugar a experiência subjetiva, que é a essência de uma emoção, e que as manifestações objetivas no comportamento e na fisiologia do corpo ou do cérebro são, na melhor das hipóteses, indicadores indiretos dessas experiências internas; além dessa, a resposta fisiológica, que se trata de reações do corpo experimentadas durante uma situação, como taquicardia ou mãos suadas; e pôr fim a resposta comportamental ou expressiva, a expressão real da emoção, como um sorriso para indicar felicidade ou um olhar baixo para indicar tristeza.

Neste sentido, existem dois tipos de comunicação: a verbal, que talvez seja o modo de comunicação mais óbvio e poderoso, a partir do compartilhamento de informações entre indivíduos usando palavras, sejam ditas ou escritas; por outro lado, a comunicação não verbal inclui expressões faciais, gestos, paralinguística, como volume, velocidade ou tom de voz, linguagem corporal, olhar, toque e aparência. Essa última pode gerar diversas interpretações, indicando a individualidade de cada pessoa envolvida e suas distintas experiências. Sendo assim, ao avaliar a comunicação não-verbal,

especificamente à linguagem corporal, é interessante atentar-se a alguns sinais que a fisionomia, em geral, possa transmitir.

3.2 EXPRESSÕES FACIAIS

A Expressão facial se trata de um dos mais importantes aspectos para eventual entendimento do ser humano. A série “Lie to Me” é um exemplo de como pode-se utilizar esse meio para identificar uma possível mentira, por meio de investigações juntamente à equipe especialista em identificar essa inverdade, a partir da leitura de gestos e microexpressões faciais de cada suspeito. Isso porque essas expressões são maneiras que o ser humano emprega para revelar suas emoções e sentimentos, de modo voluntário ou involuntário, de acordo com o jeito de cada indivíduo.

A diferença na expressão de uma pessoa que recebeu um presente agradável e, por outro lado, a de um motorista cujo carro tenha sido abatido no trânsito é perceptível, isso porque são maneiras do corpo, consequentemente, o rosto, reagir a determinada situação em que está inserido - juntamente às reações comportamentais da fisiologia de cada um.

Segundo Ekman (2003) e Plutchik (2002), essa situação possa ser explicada pelo simples fato de a face ser a região corporal com maiores mecanismos para expressão do indivíduo, ou seja, o conjunto de todos os músculos dessa área são essenciais e responsáveis pela própria revelação e demonstração das expressões de cada um, conforme a situação em que está inserido.

Em *The Expression of the Emotions in Man and Animals*, se defende a tese de que todos os humanos e até mesmo outros animais, demonstram emoções por meio de comportamentos semelhantes (DARWIN, Charles, 1872). Hoje, muitos psicólogos concordam que certas emoções são universais para todos os seres humanos, independentemente da cultura: raiva, medo, surpresa, aversão, felicidade e tristeza. Tendo isso em vista, é importante partir do pressuposto que as emoções são as principais a serem investigadas, a partir delas, pode-se analisar suas derivadas.

Isso acontece porque existem diversas maneiras de reconhecimento facial que possibilitam a identificação das expressões humanas, tais como: curvatura da boca, presença ou não de sorriso, altura e posição das sobrancelhas, franzimento da testa, direção dos olhos. Na junção de todos esses aspectos, é possível identificar o que uma pessoa pode estar sentindo, permitindo que ela expresse quaisquer emoções - por meio de ações musculares que diferenciem particularmente uma emoção da outra.

A tecnologia é um grande avanço nesse aspecto, já que muitas empresas com influência e expertise utilizam a ferramenta de identificação facial para suas respectivas finalidades - cadastro de perfil na base de clientes, desbloqueio de celular, pagamento em bancos, até mesmo identificação de suspeitos, dentre outras. Nesse sentido, para o andamento do trabalho, para haver uma maior fidelização de clientes em uma determinada empresa cujo serviço fora fornecido, é importante a análise das expressões faciais destes, com a finalidade de reconhecer seu grau de satisfação com base no atendimento que lhe foi prestado - com o respectivo consentimento do cliente e dentro de todas medidas legais.

3.3 FACS E MICROEXPRESSÕES

Para aprofundarmos sobre microexpressões faciais, é importante antes, entendermos que para ler e compreender rostos e expressões faciais adequadamente, é preciso usar procedimentos de análise científica. Com esse propósito, foi criado o Facial Action Coding System (FACS), uma metodologia científica criada pelo Dr. Paul Ekman e Wallace Friesen, no ano de 1978, para que pudessem descrever as expressões faciais humanas, e a conversão de FACS para Emotion FACS (EMFACS) baseado em pontuações universais do estado emocional para emoções básicas.

3.3.1 FACS

FACS é um sistema de análise e codificação de rosto desenvolvido por Paul Ekman e Wallace Friesen. Como o único método reconhecido e usado globalmente, o FACS não apenas reconhece e analisa todos os aspectos dos movimentos faciais, mas também decodifica a codificação de marcadores, até alcançar o EMFACS (Emotional FACS), para identificar 7 emoções básicas e

generalizadas de qualquer pessoa, em qualquer lugar do mundo (alegria, raiva, medo, tristeza, aversão, desprezo e surpresa).

Desde a sua criação, este método científico foi tecnicamente aprimorado e implementado de forma a auxiliar profissionais de diversas áreas a obter a maior precisão possível na leitura de microexpressões, pois pode identificar qualquer expressão isolando a unidade de movimentos musculares. Em termos simples, o sistema permite o uso de códigos (Unidades de Ações - AU) para leitura de microexpressões faciais. O EMFACS é um núcleo de códigos usado para realizar a interpretação das pontuações, com base em AUs.

As *Action Units*, são as Unidades de Ações, que tem como sigla o famoso AU, as Unidades de Ações são independentes de quaisquer interpretações, podendo ser utilizadas em conjunto, o que possibilita na composição do reconhecimento facial, indo do básico às mais complexas, onde se torna necessário a utilização de softwares avançados para esse trabalho de identificação. É por meio da codificação facial que animadores de filmes e desenhos animados, criadores de jogos, câmeras de segurança, interrogatórios policiais, as inserções de reconhecimento facial em smartphones, por exemplo, conseguem realizar pesquisas e melhorar a experiência do usuário.

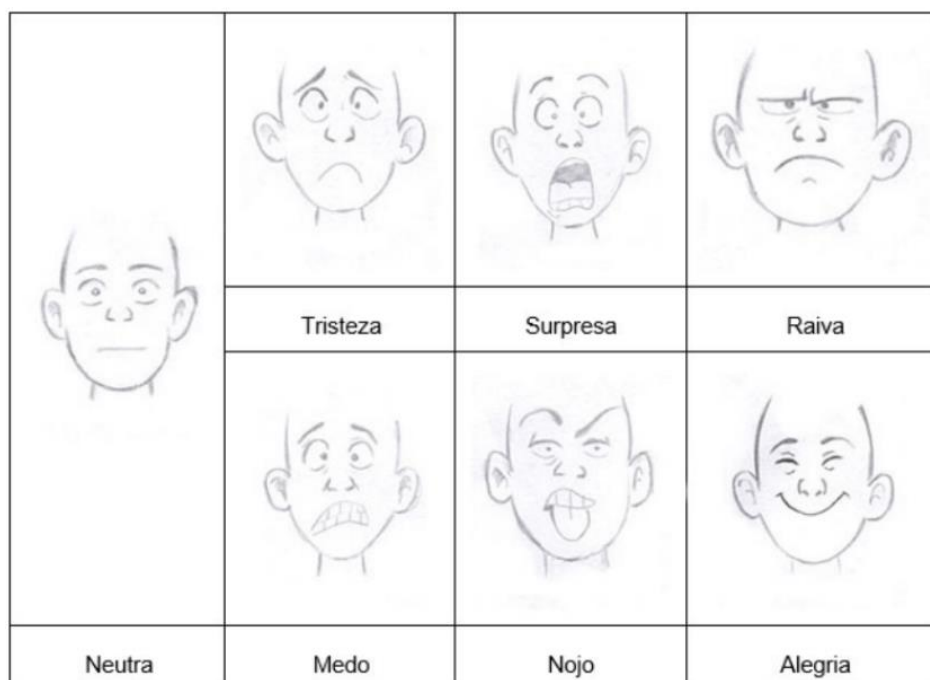
3.3.2 MICROEXPRESSÕES

As microexpressões são expressões faciais muito breves e involuntárias que os humanos fazem quando experimentam emoções. Geralmente duram de 0,5 a 4,0 segundos, não podem ser falsificadas, são caracterizadas por movimentos involuntários, de baixa intensidade (em relação aos movimentos de contração muscular), muito rápidas e estão associados ao vazamento de emoções que se tenta esconder.

As microexpressões são a chave para ler rostos, em outras palavras, o povo de qualquer país do mundo fica tão triste quanto tribos isoladas pelo mundo, que nunca viram personagens de TV ou filmes para imitar suas expressões. O Dr. Ekman chegou a descobrir que pessoas cegas de nascença fazem expressões faciais sem nunca ter visto outra pessoa fazer.

Charles Darwin desenvolveu a primeira abordagem moderna do assunto (emoções e expressões faciais) em 1872, quando publicou sua pesquisa em seu livro “The Expression of Emotions in Man and Animals”. Neste trabalho, entre muitas contribuições, ele apresenta evidências de que certas expressões faciais e os músculos que as produzem, podem estar envolvidos no desenvolvimento de certas emoções. Mas essa história de ler expressões do rosto, se popularizou com as pesquisas do Dr. Paul Ekman, que identificou em seus estudos 7 expressões faciais universais.

Figura 1 - Expressões universais por Paul Ekman e Wallace Friesen



Fonte: Bez et al (2017)

3.3.2.1 Surpresa

Embora a expressão seja breve, os movimentos faciais nos permitem observar o ambiente, mudar nosso foco para outro evento possivelmente ameaçador e reagir mais rapidamente. Seja sua causa boa ou ruim, a reação expressada pelo rosto será a mesma. As características faciais demonstradas ao sentirmos surpresa são as sobrancelhas inteiramente e pálpebras levantadas, boca aberta e pupilas dilatadas.

Figura 2 - Características que representam a microexpressão de Surpresa

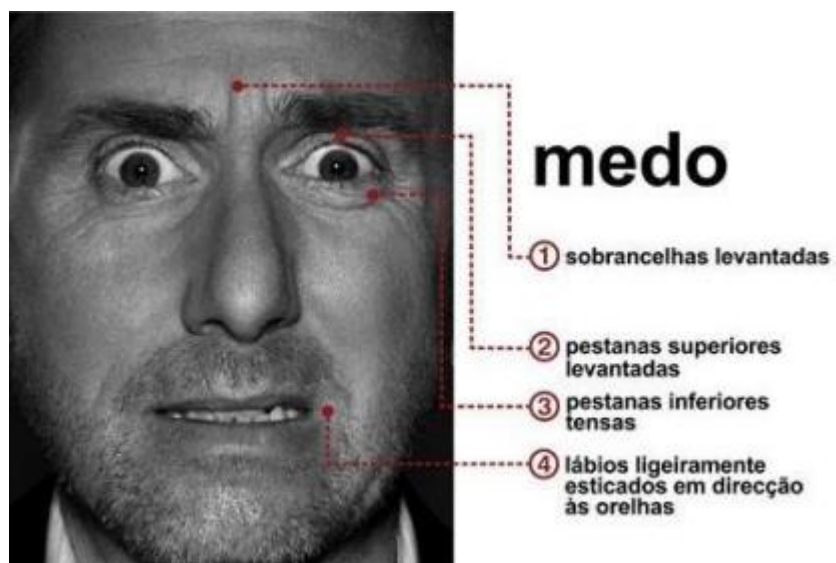


Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.3.2.2 Medo

A microexpressão do medo está intimamente ligada ao choque, cada movimento nos prepara para uma resposta de luta ou fuga. Estes movimentos faciais tiram proveito da forma que nossos corpos funcionam. Alargar nossos olhos abre nosso campo de visão, deixando entrar mais luz e permite que vejamos as ameaças ao nosso redor. O mesmo pode ser dito para nossas vias de oxigênio, abrir nossas narinas aumenta nossa ingestão de oxigênio e nos ajuda a nos preparar para fugir ou lutar. As características faciais demonstradas ao sentirmos medo, são as sobrancelhas puxadas para cima e juntas, pálpebras superiores puxadas para cima e a boca esticada.

Figura 3 - Características que representam a microexpressão de Medo

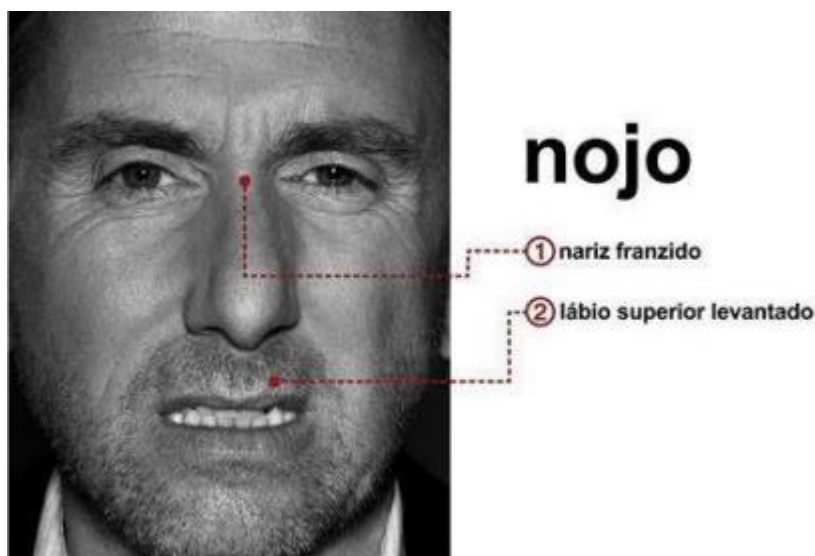


Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.3.2.3 Aversão

Quando apertamos os olhos com desgosto, nossa acuidade visual aumenta, ajudando-nos a encontrar a origem de nossa aversão. A cara de nojo, não mostra apenas nossa aversão, também funciona como proteção. Enrugar o nariz fecha a passagem nasal, dessa maneira nos protegendo de gases perigosos e apertar os olhos protege de danos. As características faciais demonstradas ao sentirmos aversão/desgosto são as sobrancelhas puxadas para baixo, nariz enrugado, lábio superior puxado para cima e os lábios inferiores soltos.

Figura 4 - Características que representam a microexpressão de Aversão

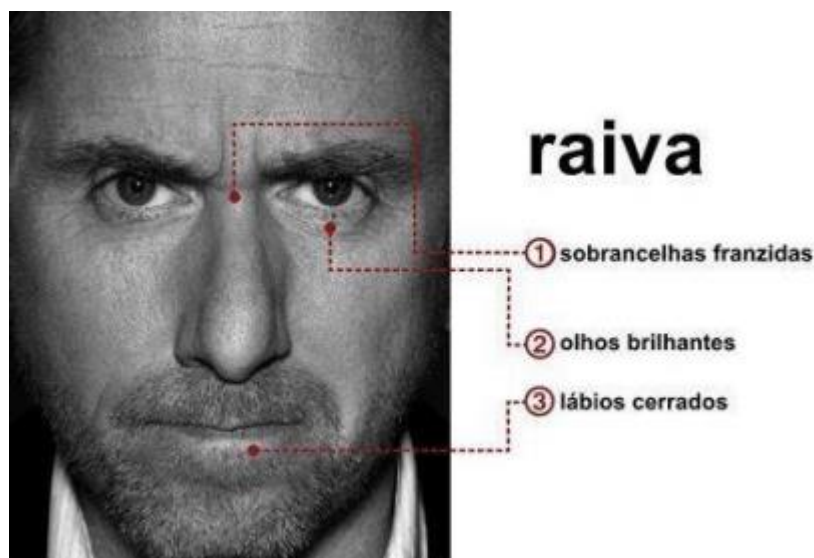


Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.3.2.4 Raiva

O rosto de raiva funciona muito bem, cada movimento desta expressão faz uma pessoa parecer mais forte, de acordo com pesquisadores. Esse rosto permite que a ameaça saiba que estamos falando sério. É uma das emoções mais poderosas e mostra o quão expressivo um rosto humano pode ser. Esse rosto serve como um aviso, seja simplesmente para intimidar ou para mostrar que um conflito começou. As características faciais demonstradas ao sentirmos raiva são as sobrancelhas puxadas para baixo, pálpebras superiores puxadas para cima, pálpebras inferiores tensionadas, margens dos lábios enroladas e os lábios podem ser apertados.

Figura 5 - Características que representam a microexpressão de Raiva



Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.3.2.5 Felicidade

Apesar da conotação amigável, muitos pesquisadores têm a convicção de que o movimento por trás do sorriso possui uma origem sinistra. Geralmente, primatas exibem seus dentes para afirmar sua superioridade e garantir seu status. Alguns pesquisadores creem que de certa forma, esta pode ser uma forma ancestral do sorriso. As características faciais demonstradas ao ficarmos felizes são os músculos à volta dos olhos apertarem, rugas perto da parte externa dos olhos, bochechas levantadas e os cantos dos lábios levantados na diagonal.

Figura 6 - Características que representam a microexpressão de Felicidade



Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.3.2.6 Tristeza

A tristeza é uma microexpressão difícil de se simular, de acordo com pesquisadores. Um dos indicadores de tristeza é a elevação da sobrancelha, que pouquíssimas pessoas podem forjar a pôr vontade própria. As características faciais demonstradas ao ficarmos tristes são os cantos internos das sobrancelhas levantados, pálpebras soltas e os cantos dos lábios puxados para baixo.

Figura 7 - Características que representam a microexpressão de Tristeza



Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.3.2.7 Desprezo

Embora o desprezo possa esconder a raiva e desconfiança, a microexpressão é singular. Podendo ser mais ou menos intensa, é a única expressão que se manifesta em apenas um lado do rosto. No seu ponto mais forte, uma sobrancelha pode abaixar enquanto a pálpebra inferior e o canto do lábio sobem do mesmo lado. Na sua forma mais secreta, o canto do lábio pode subir apenas brevemente. As características faciais demonstradas ao sentimos desprezo são os olhos neutros com o canto do lábio puxado para cima e para trás de um lado.

Figura 8 - Características que representam a microexpressão de Desprezo



Fonte: <https://www.thejournal.ie/lying-2308154-Sep2015/> (2015)

3.4 INTELIGÊNCIA ARTIFICIAL

A inteligência artificial é um subconjunto do vasto domínio da ciência da computação que se dedica a resolver problemas em tempo real que estão relacionados às atividades humanas. Embasada na inteligência humana, a inteligência artificial é a atividade de máquina que consiste no comportamento inteligente de softwares baseados em uma estrutura de algoritmos que são capazes de racionalizar, absorver conhecimento, modificar, reconhecer padrões, memorizar e comunicar de maneira absolutamente lógica e não natural com o objetivo de pensar, tomar decisões e resolver problemas práticos do mundo real.

As máquinas inteligentes estão se tornando ferramentas fundamentais que promovem a existência e constante evolução das atividades sociais cotidianas. Atualmente, a inteligência artificial pode auxiliar em diagnósticos médicos, cirurgias, resgates, vigilância, segurança, entre outros cenários e a tendência é que evolua progressivamente, resolvendo problemas cada vez mais complexos.

O Machine Learning, traduzido como Aprendizado de Máquina, e Redes Neurais são domínios fundamentais para o desenvolvimento de softwares e hardwares que utilizam inteligência artificial.

3.4.1 MACHINE LEARNING

O Aprendizado de Máquina é um dos domínios da Inteligência Artificial que promove a evolução dos algoritmos computacionais projetados para simular habilidades humanas e aprende com o ambiente ao redor, melhorando gradualmente sua precisão.

Técnicas baseadas em Aprendizado de Máquina como planejar, aprender, ouvir e falar utilizam uma ampla estrutura de algoritmos (redes neurais), ferramentas de ciência de dados para a vasta gama de tarefas de processamento e uma determinada fonte de parâmetros para analisar dados e fornecer resultados como classificações, previsões e insights com base nas informações providas pelos usuários, dessa maneira, impulsionando a tomada de decisão para determinado problema. Existem incontáveis algoritmos de Machine Learning, mas cada um é focado para solução de um problema específico.

Os algoritmos de aprendizado de máquina testam várias hipóteses imparciais com base nos recursos que educamos, dessa forma podendo realizar análises e processamentos de grande volume de dados de maneira extremamente precisa, algo intangível para o cérebro humano.

3.4.2 REDE NEURAL ARTIFICIAL

As Redes Neurais Artificiais (RNAs), também identificadas como nós interrelacionados de inteligência artificial, é um modelo computacional composto por neurônios artificiais, relativamente baseado na rede neural biológica do cérebro humano e de animais. As RNAs, buscam processar informações da

mesma maneira que o cérebro processa, identificando relações e padrões entre informações e dados integrais, associando-os e classificando-os. A rede neural é o principal componente do Machine Learning, aprendizado de máquina, pois é a estrutura de algoritmos utilizada para processamento de entrada de dados, aprendendo a realizar tarefas com base em exemplos do ambiente, sem a necessidade de programação para uma tarefa em específico. Por exemplo, no reconhecimento de imagens, as redes neurais artificiais podem aprender identificar um objeto 'X' apenas analisando parâmetros de dados denominados como 'X' e 'Y', em outras palavras analisando dados divididos por classe de reconhecimento. As redes neurais aprendem de maneira autônoma, sem ao menos ter o conhecimento de alguma característica do objeto, pois isso é realizado de maneira automática conforme vão identificando novos objetos ou novas características de objetos já identificados, de maneira similar ao processo de aprendizado por experiência do cérebro biológico humano e animal.

3.4.2.1 Neurônio Artificial

Pela razão das estruturas de RNAs terem sido desenvolvidas fundamentadas no modelo neurológico humano, os elementos computacionais ou unidades processadoras denominados como “neurônio artificial”, são modelos sintetizados dos neurônios biológicos. Os neurônios artificiais usados nas estruturas de redes neurais artificial são executados de maneira não linear, proveem saídas contínuas e realizam execuções de processamentos simples.

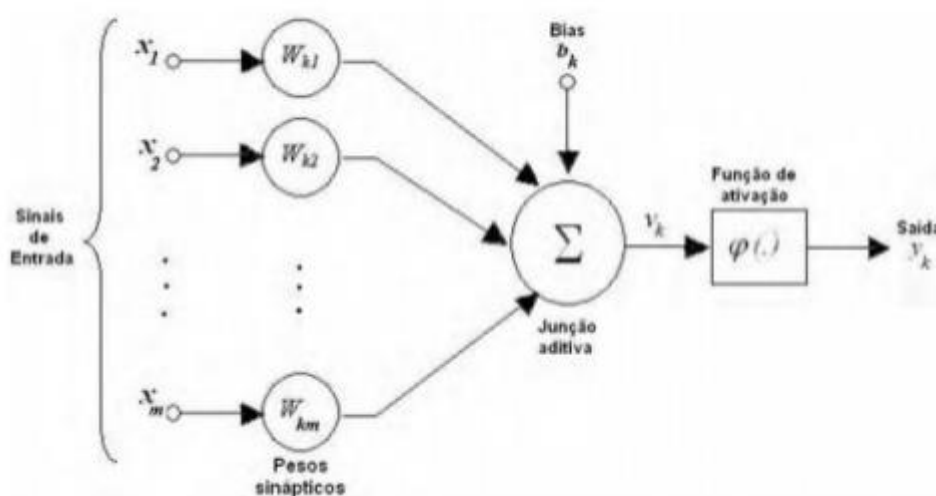
A abstração artificial de um neurônio biológico para representar o mundo real foi proposta por McCulloch e Pitts (1943). Este modelo de neurônio tenta simular as realidades biológicas que ocorrem

dentro de uma célula do sistema nervoso, é o mais simples e envolve de forma descomplicada os componentes e o funcionamento de um neurônio biológico, em outras palavras, de forma paralela e conectiva. Em termos práticos de execução, neste modelo, o neurônio receptor é estimulado de maneira variável pelos sinais de entrada (X_m), que podem ser derivados de outros neurônios, onde dentre todos as incitações, algumas podem estimular mais e outras menos. A unidade de medida do nível de estímulo do neurônio receptor é representada pelo

denominado, peso sináptico (W_{km}), que se refere força ou amplitude de uma conexão entre dois nós, onde quanto maior o peso, maior será o estímulo.

O modelo de neurônio artificial é representado abaixo, onde há elementos mencionados previamente.

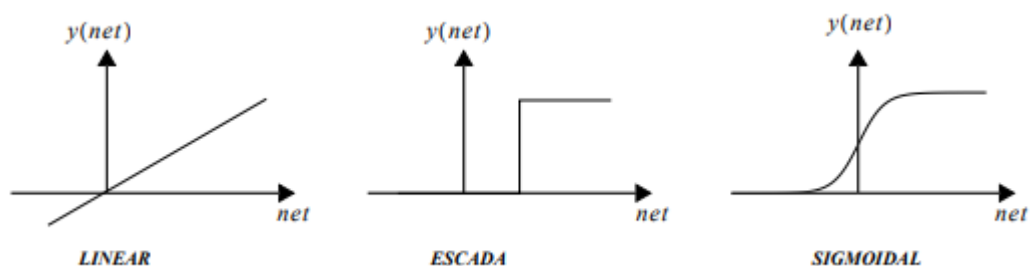
Figura 9 - Modelo de neurônio



Fonte: Adaptada de Haykin (2008)

A função de ativação orchestra o comportamento do neurônio e no modelo acima, projetado por McCulloch e Pitts (1943), não é a única forma de gerar o valor de saída do neurônio artificial, ao todo, há 3 diferentes tipos de funções de ativação.

Figura 10 - Funções de ativação



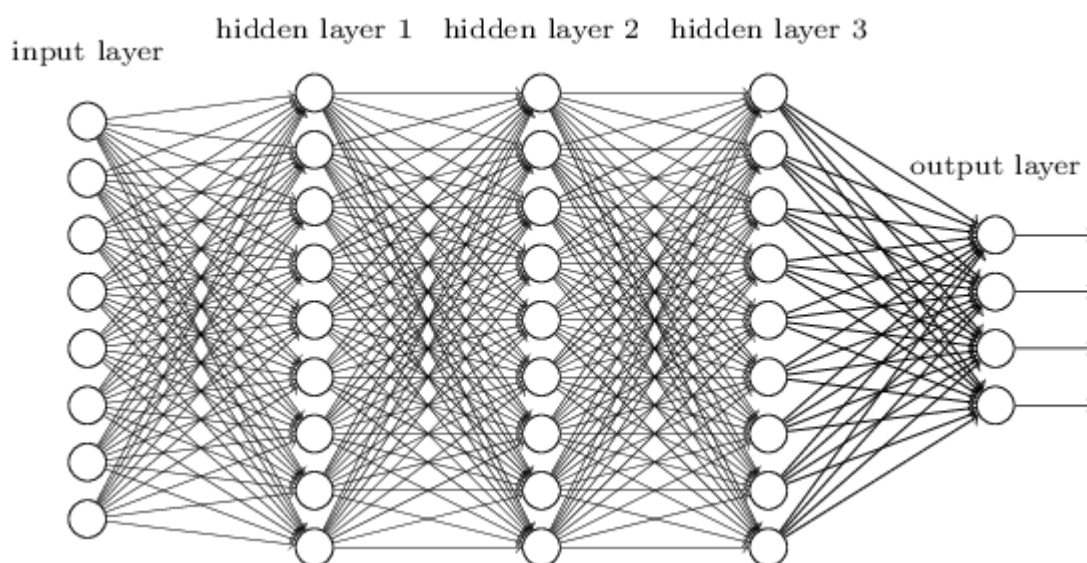
Fonte: Neural Network Toolbox (2009)

A função linear produz uma saída linear contínua, a função de escada uma saída não-linear e discreta de forma binária e a função sigmoideal produz uma saída não-linear contínua.

3.4.3 REDES NEURAIIS CONVOLUCIONAIS

Para entrarmos a fundo no tema Redes Neurais Convolucionais, é relevante que previamente tenhamos o entendimento de onde foi retirada a ideia com base biológica para a elaboração dessa arquitetura. Nos anos 60, Wiesel e Hubel realizaram uma experiência que apresentava alguns neurônios que são ativados simultaneamente quando submetidos a curvas e linhas, buscando assim conceber o reconhecimento de imagem. De uma forma simples, esse é o principal conceito de Rede Neural Convolucional: filtrar curvas, bordas e linhas e em toda camada acrescida modificar essa filtragem para uma imagem com maior complexidade.

Figura 11 - Rede Neural Profunda



Fonte: Nielsen (2015)

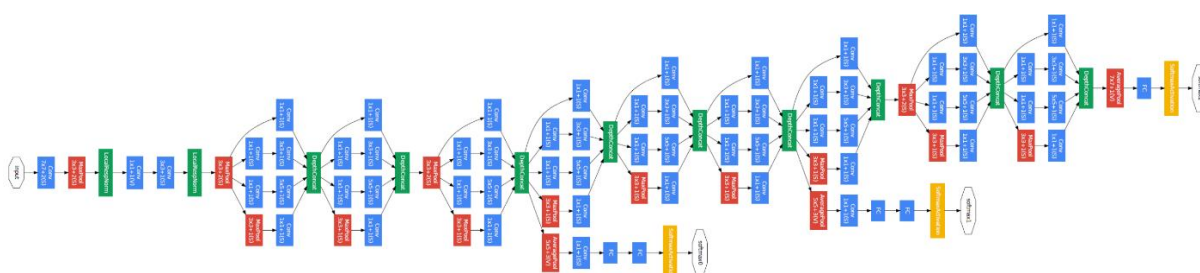
Uma antiga dificuldade na categorização de imagens se trata do próprio reconhecimento delas, e as CNN (*Convolutional Neural Network*) têm um ótimo antecedente de solução dessa dificuldade. Se tem datado que a primeira rede neural convolucional que obteve êxito teve seu desenvolvimento realizado por Yann LeCun, nos anos 90, possuindo sete camadas divididas entre convolução e

totalmente conectada. Posteriormente, as redes neurais convolucionais se desenvolveram, em versões mais complexas e robustas, como por exemplo, AlexNet, que detém 60 milhões de parâmetros mesmo possuindo somente 8 camadas (5 convoluções e 3 totalmente conectadas), enquanto o GoogleNet detém 22 camadas e 4 milhões de parâmetros.

3.4.3.1 GOOGLNET

Em 2015 foi proposta a GoogleNet, que serve de exemplar de uma arquitetura de rede neural convolucional profunda. O principal objetivo desta rede é utilizar um número mínimo de parâmetros sem perda de profundidade, módulo *Inception* proposto á isso reduz significativamente a quantidade de parâmetros de rede.

Figura 12 - GoogLeNet

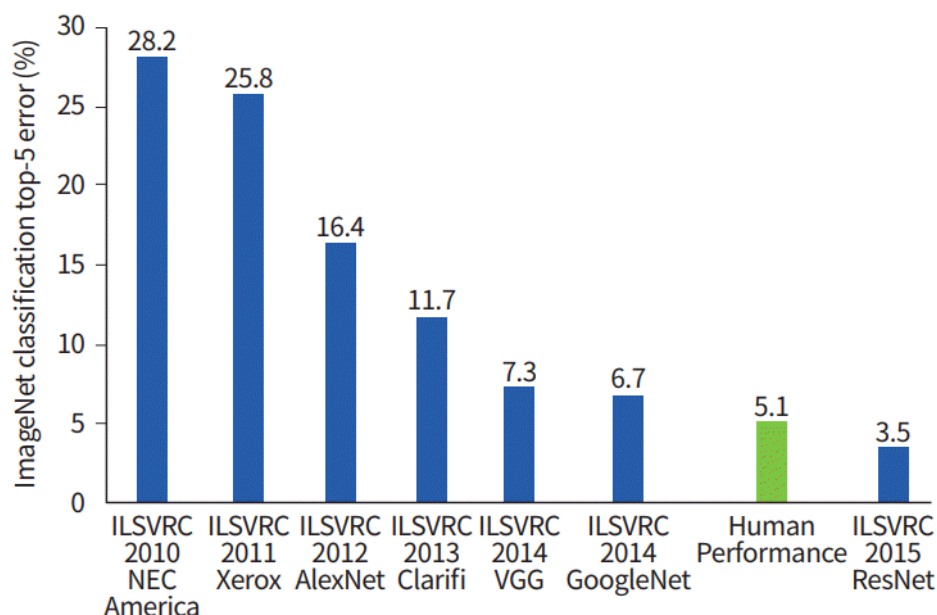


Fonte: Quoc Le & Barret Zoph (2017)

3.4.3.2 ALEXNET

AlexNet é uma arquitetura de rede neural convolucional, projetada por Alex Krizhevsky, Ilya Sutskever e Geoffrey Hinton. A Arquitetura AlexNet tem seu nome por conta de Alex Krizhevsky, um dos principais autores do artigo. AlexNet competiu e venceu o Desafio de Reconhecimento Visual em Grande Escala da ImageNet (ILSVRC) 2012 com uma taxa de erro entre os 5 primeiros de 16,4%.

Figura 13 - Top 5 classificações ImageNet



Fonte: <https://tech-pt.netlify.app/articles/pt511372/index.html>

Como o modelo tinha a necessidade de treinar 60 milhões de parâmetros (um número considerável), ele é propenso a *overfitting*. Em geral, uma das melhores maneiras é utilizar *dropout* e aumento de dados que ajuda significativamente a reduzir o *overfitting*. Portanto, a primeira e a segunda camada completamente conectada na arquitetura usam um *dropout* de 0,5 para isso. Aumentar artificialmente o número de imagens por meio do aumento de dados ajuda a expandir de uma forma dinâmica o conglomerado de dados em tempo de execução, o que ajuda o modelo a generalizar melhor.

3.4.3.3 Camadas

Pode-se desmembrar as Redes Neurais Convolucionais em dois módulos fundamentais, são eles: convolução e classificação. A fase de convolução extrai componentes de uma imagem que retratam seu conteúdo. A classificação é encarregada da função de como o próprio nome diz, classificar as informações retiradas da fase de convolução.

Uma RNC (Rede Neural Convolucional) é fragmentada em camadas, onde cada uma delas se encarrega de uma exclusiva subtarefa.

3.4.3.3.1 Camada de Convulação

Esta camada extrai e mapeia o conteúdo de uma imagem e converte em dados. O procedimento ocorre com a aplicação de filtros que possibilitam coletar elementos de sub-blocos de uma imagem.

3.4.3.3.2 Camada de Pooling

Esta camada obtém os blocos extraídos na camada anterior e descomplica a informação, reunindo os dados da imagem em um só valor e transferindo para a próxima camada.

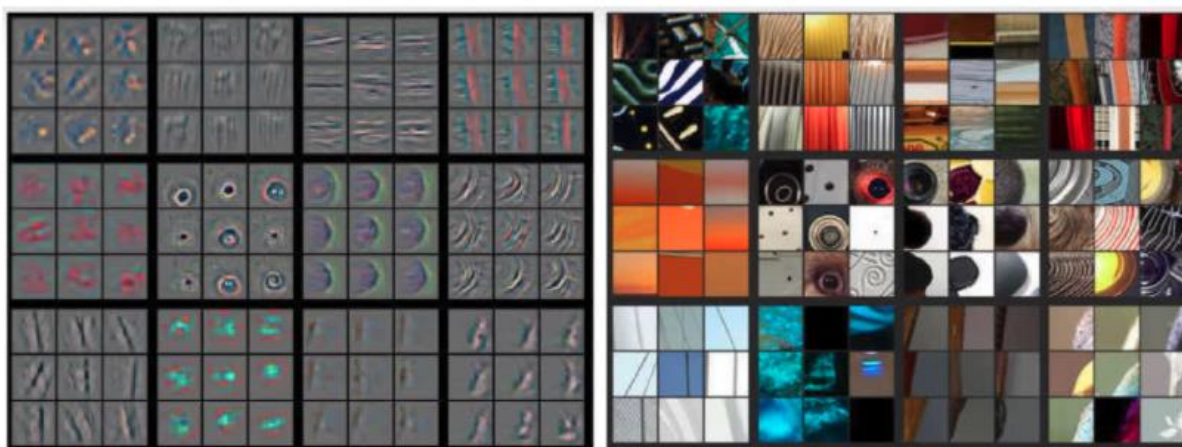
3.4.3.3.3 Camada Totalmente Conectada

Totalmente conectada é a camada onde se inicia o procedimento de classificação das informações retiradas nas demais camadas. Ela transforma o bloco em uma linha única com todos os dados fornecidos.

3.4.3.3.4 Percepção Convolucional

Durante uma aplicação de filtros dentro da camada de convolução, os filtros permitem identificar padrões na imagem, dentre elas: texturas, bordas, formas, linhas, cantos, cores, curvas e partes de objetos contidos na imagem.

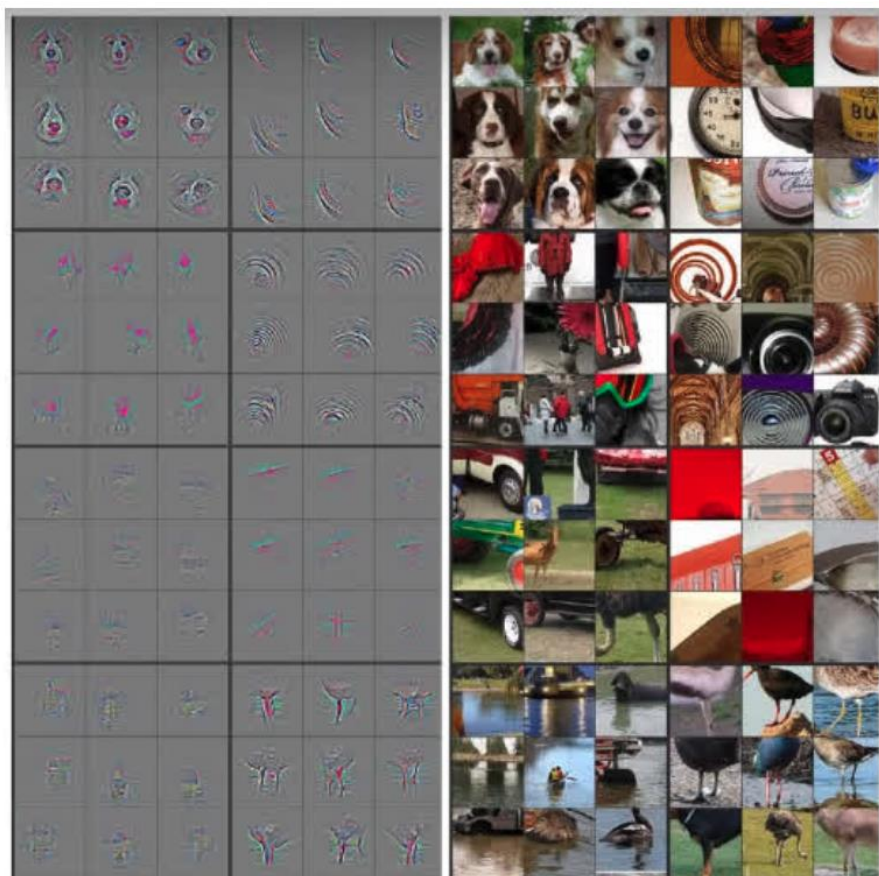
Figura 14 - Filtros de Camadas Convolucionais



Fonte: Blog Ateliware (2021)

Uma camada de rede mais profunda permite aplicar um filtro em grau aprimorado, possibilitando em camadas mais profundas a detecção de objetos como pássaros, cães e gatos em seus filtros.

Figura 15 - Filtros na Detecção de Objetos



Fonte: Blog Ateliware (2021)

3.4.3.4 Função de Ativação

Um elemento matemático incluso na infraestrutura das redes neurais decifrando problemas. Conforme a evolução da Inteligência Artificial além das inúmeras funções de ativação, novas são criadas, baseando-se em pesquisas constantemente ampliadas.

3.4.3.4.1 Função Linear

Esta função é constante, não tem dependência do valor de entrada (x). Esta função é a melhor estratégia para trabalhos simples, quando a interpretação é pretendida.

Sua fórmula consiste na seguinte função: $F(X) = AX$

3.4.3.4.2 Função Sigmóide

Esta função é constantemente consumida, é leve e diferenciável. Quando se tem muitos neurônios, a função tem variação de 0 a 1 em um molde de S. A função leva os valores (Y) para as extremidades, o que resulta em uma qualidade pretendida ao categorizar os valores em uma classe.

Sua fórmula consiste na seguinte função: $F(X) = 1 / (1 + e^{-X})$

3.4.3.4.3 Função Tanh

Esta função é similar a sigmóide, a Tanh nada mais é que uma versão escalonada da sigmóide, contudo é simétrica relacionada à origem. É contínua e diferenciável. Sua variação é de -1 a 1.

Sua fórmula consiste na seguinte função: $F(X) = 2 / (1 + e^{-2X}) - 1$

3.4.3.4.4 Função ReLU

Não é uma função linear, assim age como facilitadora na ação de solucionar erros e possuir muitas camadas de neurônios. Sua imprescindível vantagem comparada as demais funções é que ela não opera os neurônios simultaneamente, o que torna a rede eficiente e prática dentro da computação.

Sua fórmula consiste na seguinte função: $F(X) = \text{MAX}(0, X)$

3.5 OPENCV

OpenCV (Open Source Computer Vision) é uma biblioteca de programação de código aberto desenvolvida pela Intel com o objetivo de fornecer uma infraestrutura comum para aplicativos de visão computacional, acelerá-la em produtos comerciais e torná-la mais acessível. Pode ser utilizada em diversas linguagens de programação como C++, Python, Ruby e Java, são usadas para diversos tipos de análises em imagens e vídeos como tracking, reconhecimento facial, edição de fotos e vídeos, detecção e análise de textos, entre outros.

3.6 TENSORFLOW

Criado pela Google Brain, TensorFlow é uma biblioteca de código aberto para aprendizado de máquina que pode ser aplicado a uma ampla variedade de finalidades. É um sistema para criação e treinamento de máquina e muito usado para detectar correlações nos dados obtidos, buscando semelhança com a forma como humanos aprendem e raciocinam. Ele tem um ecossistema abrangente de ferramentas, e recursos aos desenvolvedores criar e implantar soluções com Machine Learning. Com suporte ao uso de CPUs multi-core, GPUs de uso geral e ASICs personalizados conhecidos como TPUs (Tensor Processing Units).

Os dados de entrada de uma aplicação utilizando o TensorFlow é um tensor, que se trata de um array multidimensional tipado. Um tensor é um vetor ou matriz com X dimensões que representa todos os tipos de dados. Um tensor pode ser originado a partir de uma extração de dados ou ser o resultado de uma operação.

3.7 KERAS

Keras é uma API de rede neural Python fácil de usar que é executada em bibliotecas de aprendizado profundo como o TensorFlow. Ele foi projetado para permitir a prototipagem rápida de modelos de aprendizado profundo, com foco na facilidade de uso, modularidade e extensibilidade. Ele suporta Redes Neurais Convolucionais (CNNs) para visão computacional e Redes Neurais Recorrentes (NNs) para processamento sequencial de dados e qualquer combinação desses dois modelos. Como o desenvolvimento da biblioteca Keras é apoiado por várias empresas importantes no ecossistema de deep learning, Keras permite que você execute o mesmo código em uma CPU ou GPU.

4 IMPLEMENTAÇÃO

4.1 MATERIAIS E MÉTODOS

Este capítulo tem por objetivo descrever os materiais e métodos utilizados para desenvolvimento, testes e implementação do projeto, além de requisitos funcionais e não-funcionais.

4.1.1 FERRAMENTAS UTILIZADAS

Em seguida, vemos a descrição e função das ferramentas que foram utilizadas no desenvolvimento do projeto.

4.1.1.1 Arduino IDE

O Arduino IDE se trata de um ambiente de desenvolvimento integrado desenvolvido pela Arduino.cc que suporta as linguagens de programação C e C++ na qual possuímos todas as ferramentas necessárias para programar placas eletrônicas, podendo escrever o código de forma satisfatória, rápida e eficiente.

No IDE, temos acesso as funções de destaque de sintaxe, correção de erros, adição de bibliotecas, monitoramento serial e envio de código para a placa, para que o programa possa rodar.

Quase todos os módulos do Arduino são compatíveis com este software que é de código aberto e de fácil instalação, estando disponível em sistemas operacionais como MacOS, Windows, Linux e roda na plataforma Java, que vem com funções e comandos embutidos que desempenham um papel essencial para depuração, edição e compilação do código no ambiente.

4.1.1.2 Microsoft Visual Studio

O Microsoft Visual Studio é uma IDE desenvolvida pela Microsoft e é usada para diversos tipos de desenvolvimento de software, como programas de computadores, sites, aplicações web e aplicações móveis em diversas linguagens de programação.

O Microsoft Visual Studio contém recursos como ferramentas para auxílio e compiladores para facilitar o processo de desenvolvimento de software como

um editor de código com suporte para *IntelliSense* e refatoração. O depurador integrado funciona tanto no nível do código quanto no nível da máquina.

Neste projeto, o Microsoft Visual Studio foi utilizado para o desenvolvimento de duas Web API's, Application Programming Interface em .Net Core 3.1, framework oriundo da linguagem de programação C#, também conhecida como CSharp.

4.1.1.3 C++

C++ é uma linguagem de programação de forma livre, estaticamente tipada, compilada, oferece suporte à programação procedural, orientada a objetos e genérica. É considerada uma linguagem de nível médio, pois compreende uma combinação de recursos de linguagem de alto, baixo nível e disponível nos sistemas operacionais MacOS, Linux e Windows.

A linguagem de programação C++ foi desenvolvido por Bjarne Stroustrup em 1979 como um aprimoramento da linguagem C e originalmente chamado "*C with Classes*", mas depois foi renomeado C++, em 1983.

Essa linguagem de programação foi utilizada para codificação do hardware ESP-32 CAM deste projeto, possibilitando o desenvolvimento das lógicas de programação para captura de imagens e comunicação com uma API, microsserviço responsável pelas lógicas funcionais da solução.

4.1.1.3 C# (CSharp)

A linguagem de programação C# (pronunciada como CSharp) é uma linguagem de programação OpenSource (código aberto), orientada a objetos, fortemente, simples, moderna, versátil, flexível, segura e eficiente, lançada pela Microsoft em 2001.

C# tem suas raízes na família de linguagens C e será imediatamente familiar aos programadores C, C++ e Java. É padronizado pela ECMA Internacional como o padrão ECMA-334 e pela ISO/IEC como o padrão ISO/IEC 23270.

4.1.1.4 .NET Core

O Framework .NET Core é uma nova versão do .NET Framework, uma plataforma de desenvolvimento gratuita, de código aberto e criada pela Microsoft. É uma estrutura de plataforma cruzada que é executada em sistemas operacionais como Windows, MacOS e Linux.

O Framework .NET Core pode ser usado para criar diferentes tipos de aplicativos, como dispositivos móveis, desktop, web, nuvem, IoT, aprendizado de máquina, microsserviços, jogos, entre outros.

O .NET Core foi escrito do zero para torná-lo um framework modular, leve, rápido e multiplataforma. Ele inclui os principais recursos necessários para executar um aplicativo .NET Core básico. Outros recursos são fornecidos como pacotes NuGet, que se pode adicionar nas aplicações conforme necessário. Dessa forma, o framework .NET Core acelera o desempenho, reduz o consumo de memória e torna-se fácil de manter, quando seguido de boas práticas de desenvolvimento de software.

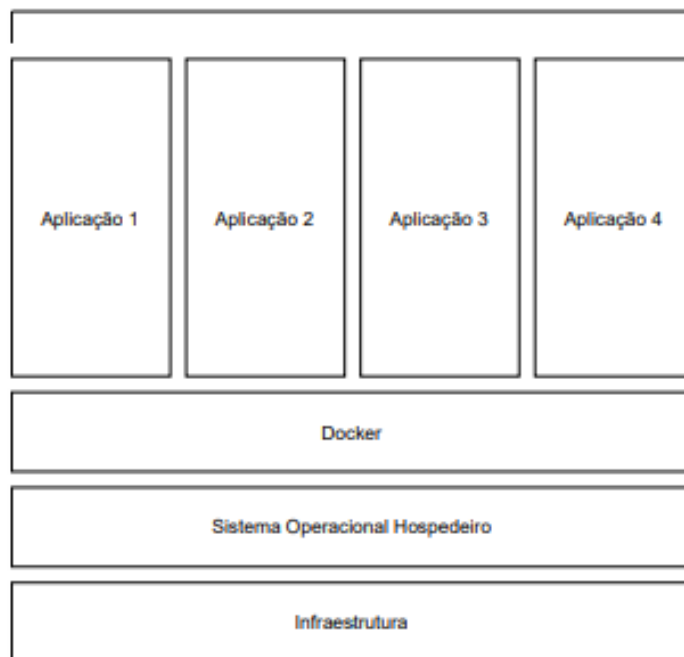
4.1.1.5 Docker

O Docker, uma ferramenta OpenSource (código aberto) lançada em 2013 é uma plataforma de software para criar aplicativos baseados em contêineres, ambientes de execução pequenos e leves que fazem uso compartilhado do kernel do sistema operacional, porém, executados isoladamente uns dos outros e neste projeto.

A containerização permite que o usuário execute aplicativos em um ambiente virtual empacotando todos os elementos necessários, como arquivos, bibliotecas e outros componentes essenciais juntos. Além disso, os contêineres desempenham um papel vital nos processos *DevOps* como compilações automatizadas de software e em implantações contínuas.

Neste projeto, o Docker foi utilizado para containerização de uma Web API em .Net Core 3.1 e implantação na plataforma *PaaS* Heroku e para geração de imagem do banco de dados MySql em ambiente local, focado para o desenvolvimento da solução.

Figura 16 - Representação Docker



Fonte: <https://www.docker.com/resources/what-container>

4.1.1.6 Heroku

Heroku é uma PaaS (*Platform as Service*), ou seja, uma plataforma como serviço na nuvem que suporta inúmeras linguagens de programação como Java, PHP, Ruby on Rails, Node, Scala, Go, Clojure e Python. Como o Heroku não é suportado pela linguagem de programação C#, o *deploy* das aplicações .NET Core é feito via contêiner, utilizando Docker, ou via *BuildPacks*. A plataforma nos permite a hospedagem de códigos e nos ajuda os usuários a não se preocuparem com a disponibilidade, escalabilidade e infraestrutura da aplicação, comumente é mais utilizada para aplicações back-end.

4.1.1.7 MySQL

O MySQL é uma das principais distribuições de SGBD (Sistema Gerenciador de Bancos de Dados) de padrão SQL disponíveis para uso comercial. É executável em vários Sistemas Operacionais, como Linux, Unix, MacOS, Windows, RTOS. É uma base de dados estável, confiável sobre seu armazenamento e gestão de dados, popular (pode-se facilmente encontrar casos de resolução de problemas na internet) e com um alto nível de segurança (dados criptografados) e pelos motivos citados, este SGBD foi utilizado para gestão de

dados neste projeto, implementado no serviço AWS RDS (Amazon Relational Database Service).

4.1.1.8 AWS RDS

O Amazon Relational Database Service ou Amazon RDS é um serviço de banco de dados relacional fornecido pela Amazon Web Services. É um serviço web com hospedagem na nuvem e projetado para simplificar a configuração, operação e dimensionamento de bancos de dados relacionais para aplicativos.

O Serviço é utilizado para gerenciar backups, patches de software, detecção e recuperação automática de falhas, pode ser habilitado backups automatizados ou pode realizar a criação manual de seus próprios snapshots de backup.

4.1.1.9 Google Colab

O Google Colaboratory, mais conhecido como Google Colab é um serviço que fornece acesso a máquinas com um grande poder de processamento para a execução de Jupyter notebooks com uma arquitetura totalmente em nuvem possuindo versão gratuita que já permite acesso a ótimas máquinas, com GPUs, discos e memória RAM inferiores aos planos pagos, mas já fornecendo uma ótima base para começar ou executar algoritmos simples.

4.1.1.10 Jupyter notebooks

Jupyter notebooks é uma tecnologia que utiliza a linguagem de programação Python, que tem sido amplamente usado devido a um de seus aspectos chave, e execução por blocos, dando a possibilidade de dividir o código a ser executado em diferentes blocos que podem serem executados de forma independente, ou seja, caso venha a ocorrer algum problema durante a execução de um bloco não é necessário executar todo o código novamente, quebrando um pouco a linearidade do algoritmo.

4.1.1.11 Kisslog

Kisslog, trata-se de um software de registro e monitoramento leve e altamente personalizável para aplicações .NET, que gera dashboards e insights

de aplicativos para facilitar a leitura e entendimento da aplicação e seu funcionamento, como a execução de solicitações de entrada e suas saídas. Uma das principais características deste software é a captura e registro automático de todas as exceções, monitoramento de tráfego HTTP, implementação orientada a objetos e registros centralizados na nuvem.

As características citadas acima foram pontos cruciais para a tomada de decisão sobre qual software de observabilidade de aplicação iríamos usar.

4.1.1.12 Swagger

As APIs são responsáveis por desempenhar um papel fundamental na comunicação entre a maioria das diferentes aplicações. O Swagger é um framework de ferramentas que facilita a descrição, consumo e visualização dos serviços de API's REST, independente da linguagem utilizada. No campo das APIs, o conceito de REST (*Representational State Transfer*) é adequado para a construção de aplicações que se comuniquem com serviços ou servidores pela WEB, utilizando o protocolo HTTP para extrair, inserir, alterar e excluir dados nesses serviços/servidores.

Optamos pela utilização da ferramenta Swagger por sua alta qualidade na documentação, tendo o excelente recurso de permitir que a documentação possa evoluir em paralelo ao desenvolvimento, dado que pode ser emulada de forma automática com base nas anotações presentes no código.

4.1.2 ARQUITETURA DE SOLUÇÃO UTILIZADA

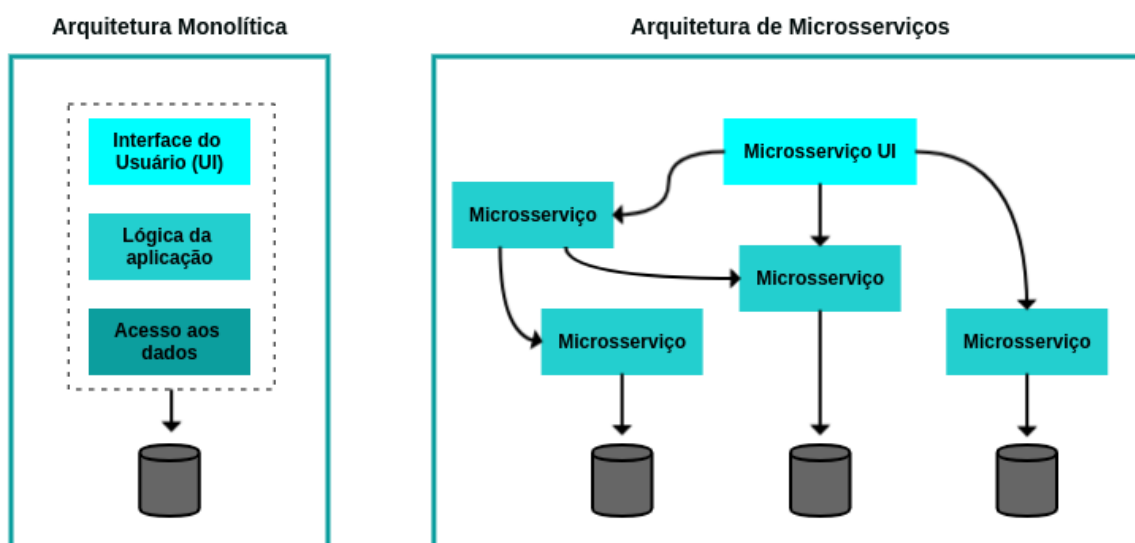
4.1.2.1 Arquitetura de Microserviços

A arquitetura de microserviços é uma abordagem para desenvolver uma única solução como um conjunto de pequenos serviços, cada um executando seu próprio processo e se comunicando com mecanismos leves, geralmente utilizando o recurso HTTP.

Esses serviços são criados com base nos domínios de negócio e construídos de maneira independente, até mesmo escritos em linguagens de programação e usando tecnologias de armazenamento de dados diferentes.

Para que a comunicação entre microserviços seja estabelecida, comumente é utilizada a tecnologia HTTP, que atua de forma síncrona ou via mensageria, que atua de maneira assíncrona.

Figura 17 - Arquitetura monolítica e microserviços



Fonte: <https://blog.4linux.com.br/microservicos-o-que-sao-onde-habitam-e-como-a-4linux-pode-te-ajudar/>

4.1.2 REQUISITOS DE SOFTWARE

Segundo Pressman (2011, p.126), os requisitos de um sistema estão entre as tarefas mais difíceis que um engenheiro de software pode enfrentar, afinal, os requisitos de um software dependem do tipo do sistema que está sendo desenvolvido, dos usuários que o software será direcionado, entre muitos outros

aspectos e cada uma dessas características dever ser definidas como funcionalidades diretas e não diretas do sistema, em outras palavras, definidas como requisitos funcionais e não-funcionais.

4.1.2.1 Requisitos Funcionais

Os requisitos funcionais são referentes ao que o sistema não pode deixar de ter, ou seja, suas principais funcionalidades. A tabela abaixo especifica os requisitos funcionais do projeto em questão.

Tabela 1 - Requisitos funcionais

Requisito	Prioridade	Descrição
Comunicação entre Arduino e API	Essencial	Desenvolvimento de lógica para comunicação entre hardware e software em ambas as partes.
Integração entre API e Inteligência Artificial	Essencial	Desenvolvimento de lógica para integração de API com a Inteligência Artificial para processamento de reconhecimento de emoções.
Persistência de dados pós processamento	Essencial	Persistir dados gerados pela Inteligência Artificial em banco de dados MySql
Disponibilização de dados por API	Essencial	Desenvolvimento de lógica de acesso à dados na API para consumo e disponibilização de dados dia rotas baseadas em arquitetura REST.
Integração de ferramentas para observabilidade	Essencial	Desenvolvimento de ferramentas como gráficos e relatórios para fácil visualização e análise de dados, como o software KissLog.
Implantação de API em ambiente Heroku	Essencial	Implantação e integração de API .Net Core 3.1, MySql e Inteligência Artificial em ambiente AWS para disponibilização ao usuário final.

Fonte: O autor

4.1.2.2 Requisitos Não-Funcionais

Conhecidos como requisitos de qualidade, os requisitos não-funcionais, são os comportamentos e estrutura do sistema como sua usabilidade, segurança,

eficiência, manutenção, arquitetura, entre outras características que não são representados através de funções.

A tabela abaixo especifica e descreve os requisitos de qualidade do projeto em questão.

Tabela 2 - Requisitos não-funcionais

Requisito	Prioridade	Descrição
Logs de informações da API	Essencial	Desenvolvimento e configuração de logs de informações da API
Logs de avisos da API	Essencial	Desenvolvimento e configuração de logs de avisos da API
Logs de erros da API	Essencial	Desenvolvimento e configuração de logs de erros da API
Segurança de Informações	Essencial	As informações de clientes, como as imagens capturadas pelo hardware, devem ser criptografadas pré-processamento e excluídas pós-processamento.
Sistema de Gerenciamento de Banco de dados	Essencial	O bando de dados a ser utilizado deve ser o MySQL.
Linguagem de programação C# e uso de .Net Core 3.1 para desenvolvimento de API	Essencial	A API deve ser desenvolvida em linguagem de programação CSharp incluindo a utilização do Framework .Net Core 3.1.
Linguagem de programação Python para desenvolvimento e treino da Inteligência Artificial.	Essencial	A Inteligência Artificial deve ser desenvolvida e testada a partir da linguagem de programação Python.
Consistência de dados	Essencial	Os dados não podem ser corrompidos ou excluídos por falha do sistema

Fonte: O autor

4.2 DESENVOLVIMENTO

4.2.1 INTELIGÊNCIA ARTIFICIAL

4.2.1.1 *Dataset*

O banco de imagens utilizado foi o FER2013, esta base originou-se de um desafio de reconhecimento facial disponibilizando 28709 imagens para treinamento, 3589 para teste e a mesma quantidade para validação, são imagens de expressões faciais de 48 por 48 pixels em tons de cinza classificadas em 7 categorias, neutro, felicidade, tristeza, raiva, medo, aversão e surpresa. Classificadas com base nos primeiros estudos de Ekman, por isso, a falta da categoria desprezo. (CARRIER; COURVILLE, 2013).

4.2.1.2 Modelo

A inteligência artificial desenvolvida neste projeto é baseada no modelo Xception, que foi originalmente proposto como uma rede neural convolucional baseada inteiramente em camadas de convolução separáveis em profundidade. Pode-se dizer que a hipótese que levou a criação do modelo Xception se trata de uma versão mais extrema da arquitetura Inception, por isso o nome Xception, que significa “Extreme Inception”.

Em resumo, a arquitetura Xception foi criada por Francois Chollet, criador da biblioteca “Keras”, para ser uma pilha linear de depthwise separable convolution layers com conexões residuais. Tornando-a uma arquitetura muito fácil de definir e modificar usando uma biblioteca de alto nível como Keras ou TensorFlow. A arquitetura Xception possui 36 camadas convolucionais que formam a base para a extração de recursos na base da rede neural.

4.2.2 HARDWARE

A Seguir, uma descrição a respeito da ferramenta hardware utilizada, tendo por objetivo descrever de forma aprofundada e detalhada suas características e detalhes.

4.2.2.1 ESP32 Cam

O Módulo ESP32-CAM é um microcontrolador fabricado pela empresa Espressif Systems que possui alta performance, baixo consumo de energia (tensão de 3.3V), capacidade de fornecer comunicação sem fio através de conexão Wireless Fidelity (WiFi) padrão 802.11 b/g/n e através do próprio sistema Bluetooth, 4MB de memória, CPU de dois núcleos, 448Kb de memória ROM, 520KBytres de memória RAM, 36 portas e dentre elas, 12 podem ser utilizadas como conversor analógico digital de 12 bits e Bluetooth. (SYSTEMS, 2019).

O ESP32 Cam é um pequeno módulo de câmera com chip ESP32-S e câmera OV2640 de 2 megapixels. Este módulo também comporta um LED de alto luminosidade para atuar como flash e uma entrada para cartões SD.

Dado sua capacidade de processamento, o módulo permitirá a captura de imagens de clientes durante atendimentos para posteriormente, processamento e resultados dos atendimentos prestados.

Figura 18 - Imagem ESP32 CAM



Fonte: <https://blog.eletrogate.com/introducao-ao-esp32-cam/>

Para configuração deste módulo, é necessário a instalação do complemento ESP32 no Arduino IDE. Todas suas funções são programadas utilizando a linguagem de programação C++, assim como a placa eletrônica do Arduino.

O uso do software citado é simples, primeiramente é necessário realizar a configuração do modelo da placa a ser utilizada, depois selecionar a porta serial virtual do dispositivo conectado ao computador e após a fase de desenvolvimento, transferir o código para o hardware por meio do botão carregar, presente no software Arduino IDE. (ARDUINO, 2015).

4.2.3 SOFTWARE

Esta seção tem como objetivo a explicação das aplicações desenvolvidas que compõem a solução como um todo. Utilizamos a arquitetura de microserviços para este projeto, onde a comunicação é estabelecida via requisições HTTP.

4.2.3.1 API Bridge

Esta Web API concentrará a maior parte das regras funcionais, afinal, será a peça centralizadora que integrará todos os outros serviços, como o hardware, inteligência artificial, banco de dados e disponibilizará endereços para consulta aos dados gerados.

A API Bridge foi desenvolvida com a linguagem de programação C# com framework de desenvolvimento .NET Core 3.1 e implantada no serviço Heroku via container, previamente citado na seção “Materiais e Métodos”.

4.2.3.1.1 Arquitetura do software

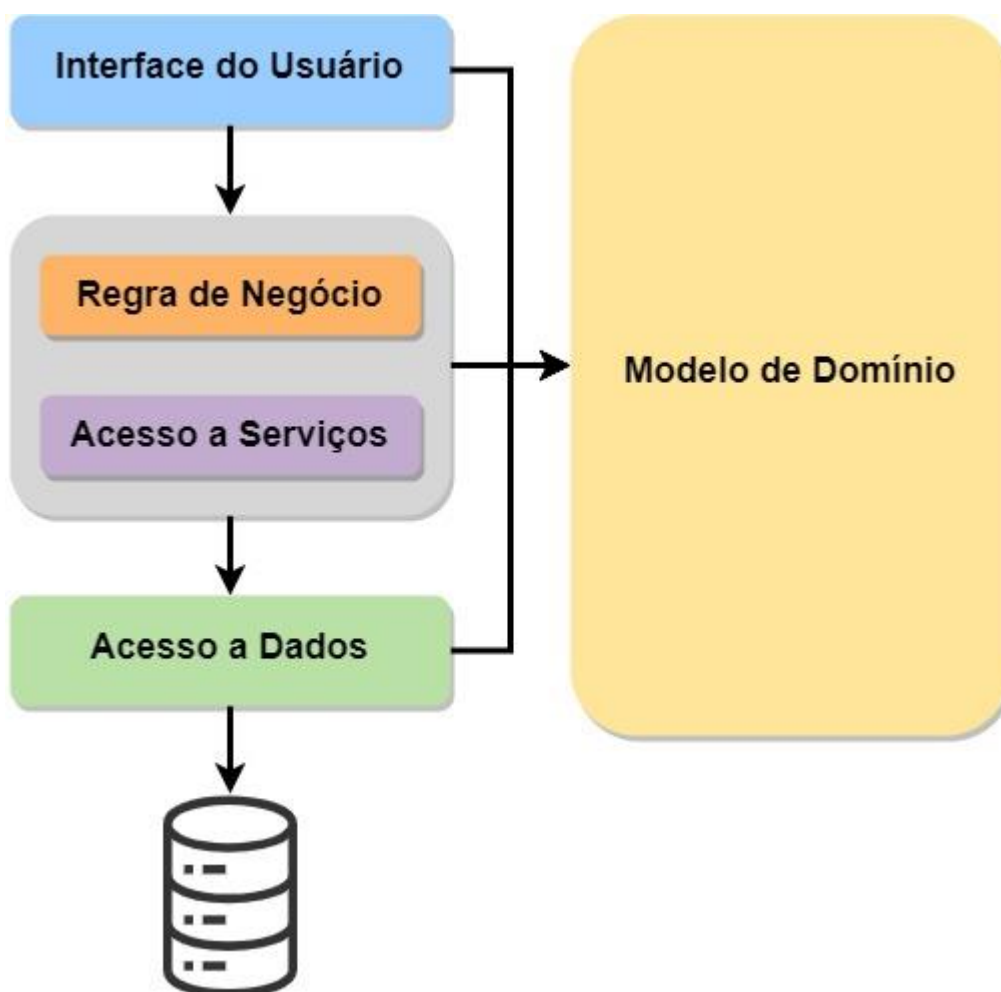
O desenvolvimento da aplicação durante o projeto teve como objetivo utilizar as tecnologias citadas na Seção “FERRAMENTAS UTILIZADAS” juntamente à uma arquitetura de *software* robusta, modular, escalável e de fácil manutenção. O *software* foi desenvolvido seguindo as boas práticas da arquitetura de N camadas, também conhecida como arquitetura de distribuidor, onde “N” representa o número de camadas, neste projeto, especificamente, camadas de apresentação, negócio, inteligência artificial e dados.

Essa arquitetura contém, além das camadas de apresentação, negócio e dados, mais camadas no servidor de aplicativos intermediando entre a interface do usuário e o banco de dados. Isso é feito para distribuir as lógicas de negócios,

até mesmo entre diferentes servidores, para que todos possam atuar individualmente.

O aplicativo baseado em arquitetura de N camadas é mais seguro do que as outras camadas de arquitetura. Até mesmo alterações ou modificações em qualquer camada podem ser feitas sem afetar as demais camadas. A qualquer momento, a lógica de negócios pode ser modificada ou o tamanho do banco de dados pode ser aumentado, sem afetar o cliente.

Figura 19 - Arquitetura de Software N camadas em API Bridge



Fonte: O autor

Dentre as camadas padrões nessa arquitetura estão:

Camada de apresentação: Essa camada, que é construída com HTML5 , folhas de estilo em cascata (CSS) e JavaScript , é implantada em um dispositivo de computação por meio de um navegador da Web ou de um aplicativo baseado

na Web. A camada de apresentação se comunica com as outras camadas por meio de chamadas de interface de programa de aplicativo (API).

Camada de negócio: Também conhecida por camada de aplicativo e camada lógica, é escrita em uma linguagem de programação como Java ou C# e contém a lógica de negócios que suporta as funções principais do aplicativo. A camada de aplicativo subjacente pode ser hospedada em servidores distribuídos na nuvem ou em um servidor interno dedicado, dependendo de quanto poder de processamento o aplicativo requer.

Camada de dados: A camada de dados consiste em um banco de dados e um programa para gerenciar o acesso de leitura e gravação a um banco de dados. Essa camada também pode ser chamada de camada de armazenamento e pode ser hospedada localmente ou na nuvem. Os sistemas de banco de dados populares para gerenciar o acesso de leitura/gravação incluem MySQL, PostgreSQL, Microsoft SQL Server e MongoDB .

4.2.3.1.2 REST

A proposta para a API tem como molde o design de arquitetura definido por Fielding (2000), ou seja, se propõe à construção de uma API REST. Desta forma, uma das definições atribuídas à construção da API é a definição de URIs e utilização dos verbos HTTP, tirando proveito do design REST para facilitar a usabilidade da API.

Figura 20 - Comandos REST e suas ações

Verbo	Objetivo	Uso	Múltiplas requisições	Cache/Bookmark
GET	Recupera itens do recurso	Link	Sim	Sim
POST	Cria um novo item no recurso	Formulário	Não	Não
PUT	Substitui itens existentes no recurso	Formulário	Sim	Não
PATCH	Atualiza um item existente no recurso	Formulário	Sim / Não	Não
DELETE	Exclui um item existente do recurso	Formulário / Link	Sim	Não

Fonte: <https://usemobile.com.br/api/>

A API pode ser vista como ponte entre a base de dados e a interface do usuário, mas que tem uma camada de validação dos dados de entrada, aplicando tanto regras de negócio quanto verificação da qualidade dos dados.

Esta camada de responsabilidade da API evita a construção de uma base de dados incorreta, além de evitar erros de consultas ou inserções inesperadas no banco de dados. A validação tem por objetivo evitar os erros de utilização da API, como inserção de dados incompletos ou incorretos na aplicação, por exemplo.

O formato de dados escolhido para ser utilizado no corpo das requisições para envio e resposta de informações foi o JSON, devido à sua simplicidade de manipulação.

A Web API disponibiliza rotas em métodos POST e GET para transferência de dados, ao todo serão dois *endpoints* disponíveis:

1. GET: Obter dados que foram gerados a partir do processamento de análise de emoções por parte da inteligência artificial e persistidos na base de dados, onde o parâmetro de busca é a identificação do colaborador e retornará todos os resultados de atendimento de um colaborador específico.

Figura 21 - Rota para obter classificações de atendimentos de colaborador

GET /analise-expressoes/classificacoes/colaborador/{funcional}

Parameters Try it out

Name	Description
funcional * required integer (\$int32) (path)	<input type="text" value="funcional"/>

Responses

Code	Description	Links
200	Success	No links
404	Not Found	No links

Media type:

Example Value | Schema

```
{
  "type": "string",
  "title": "string",
  "status": 0,
  "detail": "string",
  "instance": "string",
  "additionalProp1": "string",
  "additionalProp2": "string",
  "additionalProp3": "string"
}
```

Fonte: O autor

2. GET: Obter dados que foram gerados a partir do processamento de análise de emoções por parte da inteligência artificial e persistidos na base de dados, onde o parâmetro de busca é a identificação da empresa e retornará todos os resultados de atendimento de todos os colaboradores da organização específico.

Figura 22 - Rota para obter classificações de atendimentos de organização

GET /analise-expressoes/classificacoes/organizacao/{cnpj}

Parameters

Name	Description
cnpj * required integer(\$int32) (path)	<input type="text" value="cnpj"/>

Try it out

Responses

Code	Description	Links
200	Success	No links
404	Not Found	No links

Media type

Example Value | Schema

```
{
  "type": "string",
  "title": "string",
  "status": 0,
  "detail": "string",
  "instance": "string",
  "additionalProp1": "string",
  "additionalProp2": "string",
  "additionalProp3": "string"
}
```

Fonte: O autor

3. POST: Envio de dados e imagens do hardware para a Web API com finalidade de processamento da análise das expressões do cliente durante atendimento.

Figura 23 - Rota para enviar imagens de atendimentos

POST

/analise-expressoes/classificacoes

Try it out

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "funcional_colaborador": 0,
  "imagem": "string"
}
```

Responses

Code	Description	Links
201	Created	No links
404	Not Found	No links

Media type

application/json

Example Value | Schema

```
{
  "type": "string",
  "title": "string",
  "status": 0,
  "detail": "string",
  "instance": "string",
  "additionalProp1": "string",
  "additionalProp2": "string",
  "additionalProp3": "string"
}
```

409 Conflict | No links |

Media type

application/json

Example Value | Schema

```
{
  "type": "string",
  "title": "string",
  "status": 0,
  "detail": "string",
  "instance": "string",
  "additionalProp1": "string",
  "additionalProp2": "string",
  "additionalProp3": "string"
}
```

Fonte: O autor

4.2.3.2 API IA

Esta API tem a função de execução e retorno da inteligência artificial e foi implantada de forma apartada pelo motivo de capacidade, consequentemente melhorando a performance da aplicação.

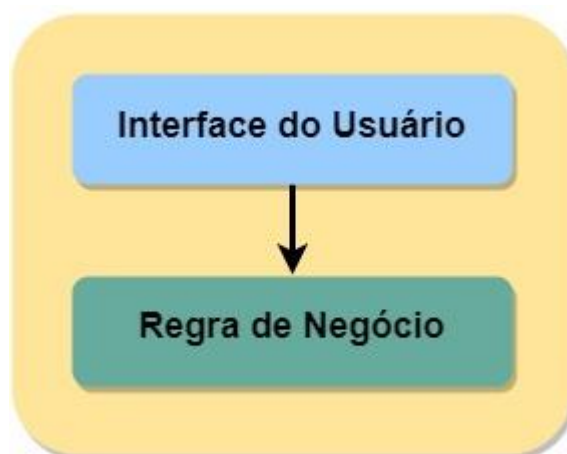
Dado que a aplicação terá uma máquina dedicada apenas para o recurso de inteligência artificial, a capacidade física da instância, como características de potência de CPU e memória RAM principalmente, estarão totalmente disponíveis para o processamento de reconhecimento facial e reconhecimento de emoções, contudo resultando em um tempo melhor de processamento do fluxo arquitetural por um todo.

A API IA foi desenvolvida com a linguagem de programação C# com framework de desenvolvimento .NET Core 3.1 juntamente as bibliotecas Keras e EMGU.cv para possibilitar a integração e execução do modelo de inteligência artificial. Esta API foi implantada no serviço Heroku via container, previamente citado na seção “Materiais e Métodos”.

4.2.3.2.1 *Arquitetura do software*

A arquitetura de software adotada para essa API é a camada única, pois se trata de uma aplicação simples e que exige utilização efetiva de recursos de hardware. É uma arquitetura útil para gerir e manter, além de ser econômica.

Figura 24- Arquitetura de Software camada única em API IA



Fonte: O autor

4.2.3.2.1 *REST*

Esta API também utiliza a arquitetura REST para disponibilizar o *endpoint* para processamento de IA:

1. POST: Iniciar processo de reconhecimento facial e reconhecimento de microexpressões baseado no conteúdo enviado no corpo da requisição, que deve ser a imagem no formato Base64.

Figura 25 - Rota para análise em API IA

POST /IaEngine/Analyze

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{  "image": "string"} 
```

Responses

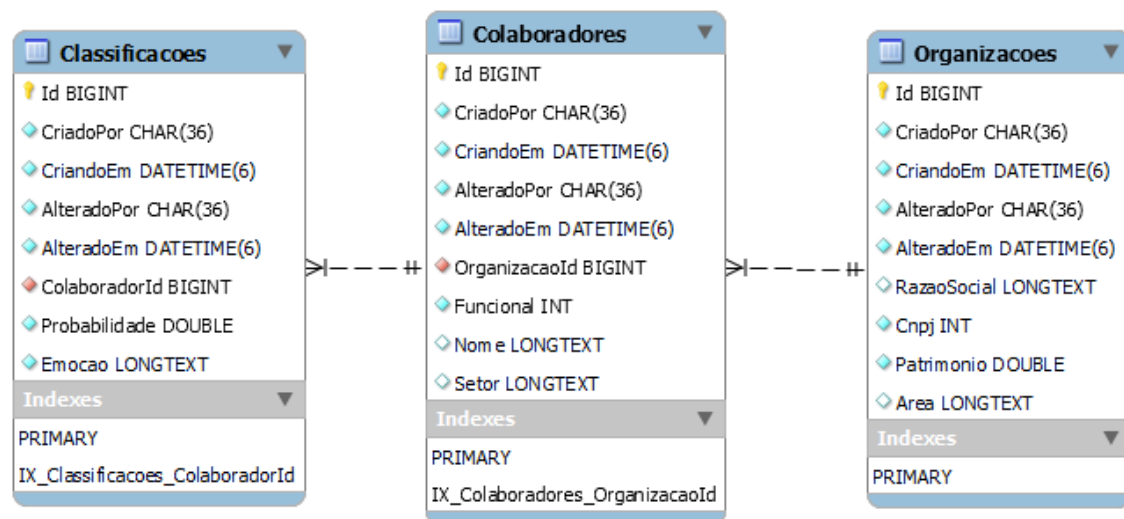
Code	Description	Links
200	Success	No links

Fonte: O autor

4.3.4 BANCO DE DADOS

O banco de dados definido para o projeto, MySql, possui a capacidade de armazenar a hierarquia e relacionamento dos elementos. Bancos relacionais possuem a capacidade de relacionar tabelas entre si através de índices, possibilitando a correlação entre emoção, colaborador e empresa. Isto, em conjunto com a capacidade de inserir tipos de dados numéricos e caracteres encadeados, que satisfaz para o objetivo.

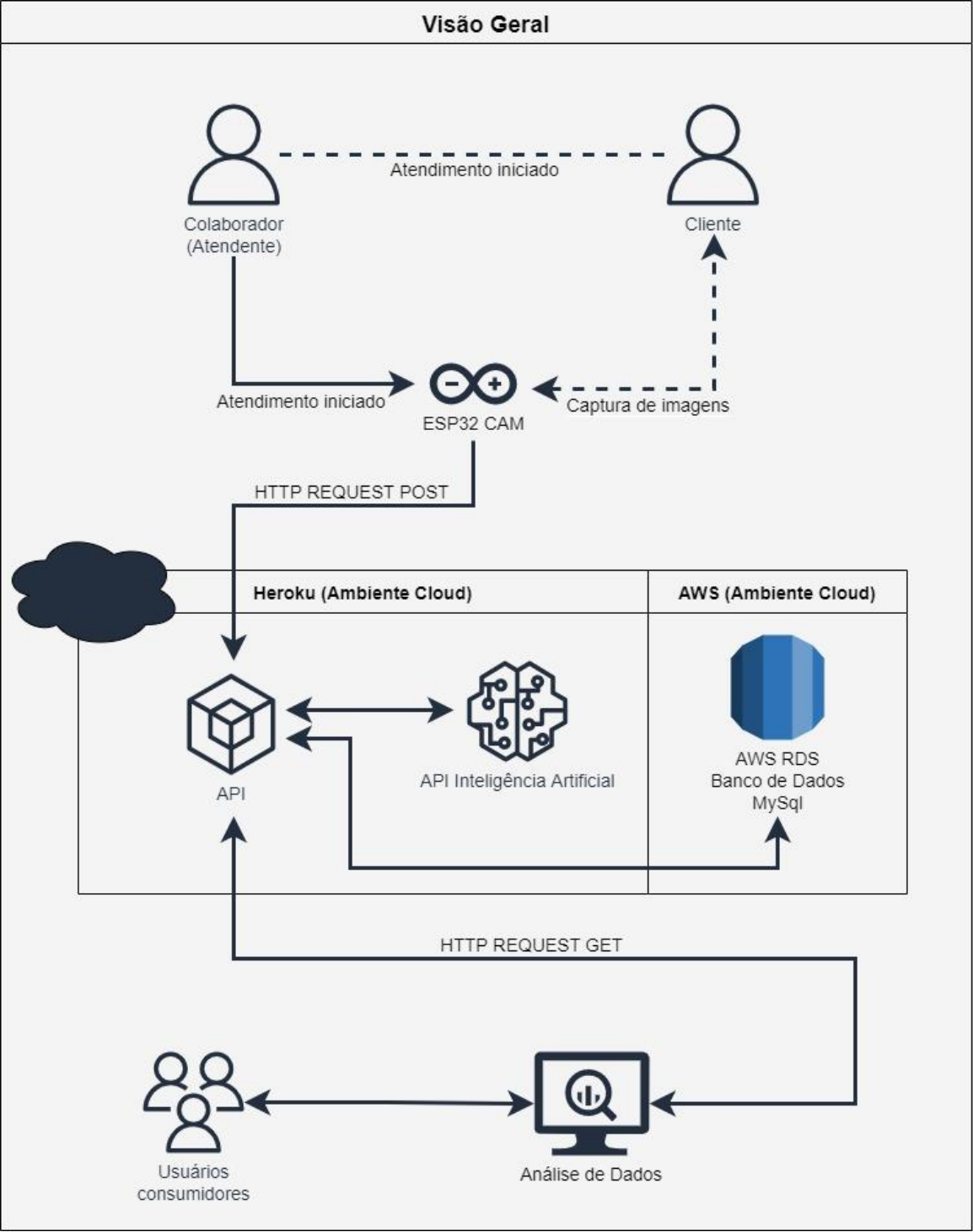
Figura 26 - Modelagem do Bando de Dados



Fonte: O autor

4.2.4 DIAGRAMA DE INTEGRAÇÃO

Figura 27 - Diagrama de integração de componentes do projeto



Fonte: O Autor

4.2.4.1 Usabilidade

O Projeto consiste em uma API Bridge, que atua como direcionador, integrada a um hardware que possui um botão físico para iniciar e finalizar o processo de captura de imagens do consumidor. Ao acionar o botão, o responsável dará início ao processo, em que a câmera iniciará com a captura de imagens, fará a conversão para Base64 e as enviará para a API, onde contém as regras e integrações que possibilita o processamento pela Inteligência Artificial, por fim, persistindo os dados retornados da análise em um banco de dados MySQL.

Temos como objetivo principal a melhoria de atendimento para a evolução das empresas e para chegarmos neste resultado precisamos trabalhar com a UX (*User Experience*) e UI (*User Interface*), que nada mais é do que a experiência do usuário, respectivamente.

Na atividade, a UX é a relação entre usuário e produto ou serviço. Sua principal função é atender que todas as etapas de interação do usuário sejam fáceis e dinâmicas, elas devem impactar positivamente os usuários e colaborarem com os objetivos do negócio. Já a UI deve ser uma interface funcional e usual, tornando-se a intermediária visual ou não, entre o usuário e a máquina.

Baseando-se em nosso projeto, utilizamos o usuário no centro de decisões do produto, com isso traremos apenas simples ações de botões que irão capturar as expressões de forma descomplicada e automática, sem gerar frustrações, isso porque normalmente as avaliações de atendimento são realizadas via formulários e são cansativas e pouco simplificadas, o que pode gerar descontentamento com a interface.

Com base em expressões faciais podemos medir a qualidade de um atendimento, um dos aspectos essenciais para o sucesso de uma corporação. Propõe-se que a empresa utilize os dados fornecidos por nossa base de dados para montar *dashboards* e *insights* para análise interna de qualidade durante os atendimentos realizados em seus estabelecimentos.

4.2.4.2 Proteção de dados

Os debates sobre como os dados dos usuários serão tratados e autorização de uso dos dados para análise são pontos chave, sendo assim, o consumidor precisa aceitar o uso de sua imagem para que o atendimento seja iniciado, caso recuse o processo não será realizado.

Considerando a Lei Geral de Proteção Dados (LGPD), uma vez que as imagens foram capturadas é de nossa responsabilidade garantir a segurança e integridade, em razão disso a criptografia será utilizada em todas as comunicações entre serviços. As imagens capturadas serão armazenadas somente até serem tratadas e analisadas pela API IA, logo após serão excluídas para garantir a confiabilidade.

4.2.5 REPOSITÓRIOS

O repositório com o código fonte do projeto (ESP-32 CAM, API Bridge, API IA, Inteligência Artificial e banco de dados) está disponível no endereço <https://github.com/pedrooplx/tcc>.

Figura 28 - QR Code para acesso ao repositório



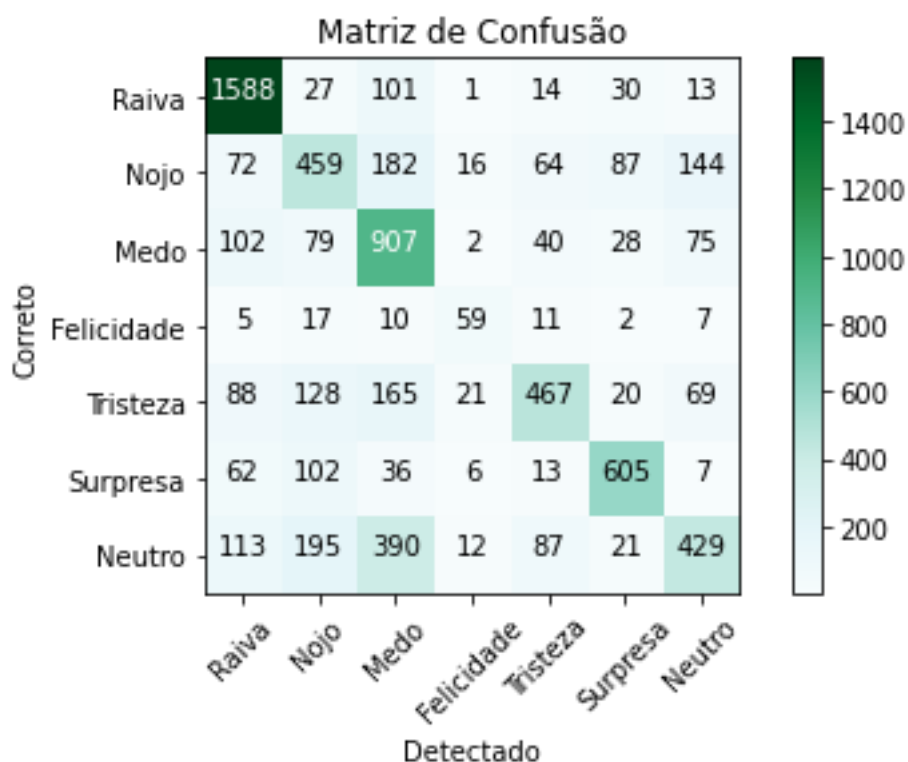
Fonte: O autor

4.3 RESULTADOS

Após a fase de aprendizado de máquina, a rede neural encontra-se apta para realizar suas classificações e previsões, com isso construímos uma matriz de confusão para identificarmos a proporção de acertos da IA após sua análise.

Na tabela, as microexpressões da esquerda são as emoções expressas nas imagens utilizadas para a identificação e as microexpressões na parte inferior são as emoções detectadas através do processamento das RNAs.

Figura 29 - Desempenho e Classificação do Aprendizado de Máquina



Fonte: O autor

Durante a fase de treinamento, o processo utiliza as imagens separadamente para testar cada passo dado, e através deste teste efetuamos o cálculo para acurácia do modelo, que é a proporção total de acertos e obteve-se o valor de 65% de acurácia.

Figura 30 - Definição de *callbacks* e montagem de logs

```
log_file_path = base_path + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)

early_stop = EarlyStopping(monitor='val_accuracy', patience=patience)
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.1, patience=int(patience/4), verbose=verbose)

trained_models_path = base_path + '_mini_xception'
model_names = trained_models_path + '.{epoch:02d}_{val_accuracy:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(filepath=model_names, monitor='val_accuracy', verbose=verbose, save_best_only=True)

callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]
```

Fonte: O autor

Na Figura abaixo, temos a seção de código que realiza o treino da rede neural. O TensorFlow usa o método `model.fit()`, responsável por treinar o modelo criado nas etapas anteriores. Os *datasets* de treinamento e as *labels* são

passados para a função, lembrando que o TensorFlow só aceita dados dos tipos de array do Numpy ou dos próprios tensores do TensorFlow. É enviado o número de amostras por atualização de gradiente e o número máximo de épocas para as quais o algoritmo deve treinar. Os dados de validação do modelo também serão passados para verificar se ele está de fato aperfeiçoando seu aprendizado.

Figura 31 - Treinamento modelo de rede neural

```
model.fit_generator(data_generator.flow(xtrain,
                                       ytrain,
                                       batch_size),
                  steps_per_epoch=len(xtrain)/batch_size,
                  epochs=epochs,
                  verbose=verbose,
                  callbacks=callbacks,
                  validation_data=(xtest, ytest))
```

Fonte: O autor

Abaixo, há exemplos de análises realizadas pela inteligência artificial sobre as imagens apresentadas a ela. Após todo o processo de aprendizado de máquina a qual foi submetida, esses foram os resultados obtidos:

Figura 32 - Face triste



Fonte: FER2013

Na figura 32, a microexpressão reconhecida foi a de “tristeza” com probabilidade de 52,08%.

Figura 33 - Face assustada



Fonte: O autor

Na figura 33, a microexpressão reconhecida foi a de “assustado” com probabilidade de 84,94%.

Figura 34 - Face feliz



Fonte: O autor

Na figura 34, a microexpressão reconhecida foi a de “feliz” com probabilidade de 76,20%.

4.4 TRABALHOS FUTUROS

O presente trabalho foi restrito em avaliar atendimentos presenciais, no entanto, atendimentos remotos não foram considerados. Recomenda-se o aprofundamento dos estudos quanto a esta funcionalidade.

Outro parâmetro a ser alterado é o hardware utilizado, a escolha de um hardware mais potente poderia aumentar qualidade de imagem e armazenamento em memória do produto.

Nessa mesma direção é desejável a implementação de uma interface mais elaborada, com um botão via software para acionamento, assim possibilitando uma interação mais convidativa aos usuários.

Propõe-se também a implementação de análises obstruídas, ou seja, trazer um tratamento especial na hora de analisar faces que apresentam barba, máscara ou outros adereços que podem prejudicar a análise.

A biometria facial seria a continuação desejada quando falamos em identificar emoções de múltiplos clientes podendo separar as emoções apresentadas individualmente para cada pessoa observada, assim abrangendo eventos mais complexos, como uma reunião de negócios.

5 CONSIDERAÇÕES FINAIS

Dado o exposto, concluímos que a Inteligência Artificial trouxe inúmeras possibilidades de soluções que anteriormente não eram possíveis. As máquinas inteligentes estão se tornando ferramentas fundamentais que promovem a existência e a constante evolução das atividades cotidianas. Ela auxilia em diagnósticos médicos, resgates, segurança, e outros cenários e a tendência é que evolua progressivamente, resolvendo problemas cada vez mais complexos. Uma de suas capacidades é a análise de expressões baseadas no reconhecimento facial, abrindo um leque de oportunidades e melhorias para processos empresariais, como o atendimento, um dos pilares mais importantes para a fidelização de clientes.

Tendo em vista essa oportunidade juntamente à finalidade de analisar os procedimentos e eficácia dos serviços prestados nas empresas, o trabalho apresentou a proposta de desenvolvimento de um protótipo que analisa a comunicação não-verbal transmitida pela face de consumidores durante determinado atendimento, retornando o percentual de presença de cada uma das emoções encontradas e disponibilizando esses dados como fonte para análise de qualidade da empresa e avaliação de seus colaboradores.

REFERÊNCIAS BIBLIOGRÁFICAS

Ekman, Paul. **A linguagem das emoções**: revolucionando sua comunicação e seus relacionamentos reconhecendo todas as expressões das pessoas ao redor. São Paulo: Lua de Papel, 2003.

Ferreira, Caio Henrique. **Base de dados CICEM: Microexpressões em Faces Humanas**. São Paulo: CICEM, 2017.

Hatfield, Elaine; Cacioppo, John T; Rapson, Richard L. **Emotional Contagion**. United Kingdom: Cambridge University Press, 1993.

Ekman, Paul; Rosenberg, E. L. (Eds.). **What the face reveals**: basic and applied studies of spontaneous expression using the facial action coding system (FACS). New York: Oxford University Press, 2005.

Ferreira, Caio Henrique. **As Faces das Emoções Básicas**. São Paulo: CICEM, 2017-2018.

Ekman, Paul; Friesen, W. V. Hager, J. C. **The Facial Action Coding System**. Salt Lake City, UT: research Nexus ebook, 2002.

Boulogne, Duchenne de. **The mechanism of human facial expression**. Nova York/EUA: Universidade de Cambridge, 1990.

Ekman, Paul. **A linguagem das emoções**. São Paulo/SP: Leya, 2011.

Ferreira, Caio Henrique; Magalhães, A. Freitas. **Facial Action Coding System 2.0: Guia F-M FACS 2.0**. São Paulo: F-MGB Lab, 2017.

Grazer, B. Lie to me, **Television Series**. California: 20th Century Fox Television, 2009-2011.

Magalhães, A. Freitas. **O código de Ekman: o cérebro, a face e a emoção**. Porto/PT: Escrytos, 2011.

Bom atendimento é mais importante para consumidor do que qualidade, revela pesquisa. **Revista Pequenas Empresas & Grandes Negócios**, 2009.

VIANA, Isabel. **Comunicação não verbal e expressão facial das emoções básicas**. Quinta de Prados/PT: Revista de Letras, nº13, 2014.

Ekman, Paul. **Would you lie to me?** EUA: Artigo publicado no jornal The Guardian, 2003.

Novaes, Felipe C. **O legado de Darwin para a compreensão das emoções**. Instituto Brasileiro de Linguagem Emocional, 2022.

Pain, Giorglli Soares. **Análise das microexpressões faciais como prova inominada**. Londrina/PR: Instituto de Pós-Graduação – IPOG, 2019.

COVIELLO, L. SOHN; Y. KRAMER; A. MARLOW; C. FRANCESCHETTI, M. **Detecting Emotional Contagion in Massive Social Networks**. *Journals Plos One*, 2014.

Jabr, Ferris. **The evolution of emotion: Charles Darwin's little-known psychology experiment**. *United States: Scientific American*, 2010.

CHOI, A. A.; MELO, B. C.; M. KHOOSHABEH, C. P.; WOOD, W; GRATCH, C. J. **Physiological evidence for a dual process model of the social effects of emotion in computers**. *International Journal of Human-Computer Studies*, 2015.

Tran, V. **The influence of emotions on decision-making processes in management teams**. Geneva, Switzerland: Université de Genève, 2004.

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. **Redes Neurais Artificiais: para engenharia e ciências aplicadas**. 2ª edição. ed. rev. e aum. São Paulo: Artliber Editora Ltda., 2016.

HAYKIN, S. S. **Redes neurais artificiais: princípio e prática**. 2a Edição, Bookman, São Paulo, Brasil, 2000.

FURTADO, Maria Inês Vasconcellos. **Redes Neurais Artificiais: Uma Abordagem Para Sala de Aula**. Ponta Grossa (PR): Atena Editora, 2019. Disponível em: <https://www.atenaeditora.com.br/wp-content/uploads/2019/05/e-book-RedesNeurais-Artificiais-uma-Abordagem-para-Sala-de-Aula.pdf>.

Roger, PRESSMAN. **Engenharia de Software: uma abordagem profissional**. AMGH, 2011. Disponível em: <https://books.google.com.br/books?id=eRIOuQAACAAJ>.

Bjarne Stroustrup (1994). **The Design and Evolution of C++**

Andrei Alexandrescu e Herb Sutter (2004). **C++ Design and Coding Standards: Rules and Guidelines for Writing Programs**.

COURVILLE, Aaron. **Challenges in Representation Learning: Facial Expression Recognition Challenge**. 2013. Disponível em: <https://www.kaggle.com/c/challenges-in-representation-learning-facialexpression-recognition-challenge>. Acesso em: 25/04/2022

EKMAN, P. **Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life**. Henry Holt and Company, 2003.

ABADI, Martín, **TensorFlow: A System for Large-Scale Machine Learning**. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, 2016. Disponível em: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>.

OPENCV TEAM. **OpenCV: about.** About. 2020. Disponível em: <https://opencv.org/>. Acesso em: 27/04/2022

BEZ, M. **Reconhecimento Facial e Micro Expressões.** 3. ed. São Paulo, Murof, 2017.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning.** MIT Press: 2016.

KERAS. How convolutional neural networks see the world, 2016. Disponível em <https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html>. Acesso em 28/04/2022

DATARISK. O que é Matriz de Confusão. Disponível em: <https://ajuda.datarisk.io/knowledge/o-que-%C3%A9-matriz-de-confus%C3%A3o>. Acesso em 27/04/2022

Passos, Bianka. **O mundo do ponto de vista das Redes Neurais Convolucionais.** Blog Ateliware. Disponível em: <https://ateliware.com/blog/redes-neurais-convolucionais>. Acesso em: 24/04/2022

Chassignol, M., Khoroshavin, A., Klimova, A., & Bilyatdinova, A. **Tendências da Inteligência Artificial na educação: uma visão narrativa.** Procedia Computer Science, 2018

Deep Learning Book. Capítulo 8 - Função de Ativação. Disponível em: <https://bityli.com/CETQsx>

HOCKENBURY, Don; HOCKENBURY, Sandra E. **Discovering Psychology.** 1990

DARWIN, Charles, **The Expression of the Emotions in Man and Animals.** 1872

AELA, Editorial. **Qual a Diferença Entre UX e UI Design? – Entenda De Uma Vez.** Disponível em: <https://aelaschool.com/experienciadousuario/qual-a-diferenca-entre-ux-e-ui-design-entenda-de-uma-vez/>. Acesso em: 17 junho 2022.

BASSO, Bianca. **Inteligência Artificial Para Reconhecimento De Emoções Em Um Ambiente Empresarial.** 17th International Conference on Information Systems and Technology Management. Disponível em: <http://contecsi.submissao.com.br/arquivos/6526.pdf>. Acesso em: 19 jun. 2022.

SYSTEMS, E. ESP32 Series datasheet. 2019.

ARDUINO. 2005, 2015, 2020. Disponível em: <http://arduino.cc/>

Fielding, Roy Thomas: Architectural styles and the design of network-based software architectures, 2000. https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.

CHOLLET, François. Xception: **Deep learning with depthwise separable convolutions**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. p. 1251-1258.