FIEL – FACULDADES INTEGRADAS EINSTEIN DE LIMEIRA

GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Júlia Resende Fernandes

Larissa Ferreira Bertoli

GATUNO: Câmera de segurança inteligente de baixo custo

LIMEIRA, SP

2024

Júlia Resende Fernandes

Larissa Ferreira Bertoli

GATUNO: Câmera de segurança inteligente de baixo custo

Monografia apresentada às Faculdades Integradas Einstein de Limeira – FIEL, como exigência parcial para obtenção do grau de tecnólogo em análise e desenvolvimento de sistemas.

Orientador: Prof. Esp. Jonas Henrique Ferreira

LIMEIRA, SP

2024

FIEL – FACULDADES INTEGRADAS EINSTEIN DE LIMEIRA

TERMO DE APROVAÇÃO

GATUNO: Câmera de segurança inteligente de baixo custo

Júlia Resende Fernandes

Larissa Ferreira Bertoli

Orientador: Prof. Esp. Jonas Henrique Ferreira

Banca examinadora:

Professor Esp. Johanny Tetzner de Souza

Professor Esp. Jonas Henrique Ferreira

Professor Dr. Pedro Ivo Garcia Nunes

Data de aprovação: 20/06/2024

*Dedicamos este trabalho à nossas famílias e amigos que nos apoiaram durante toda nossa trajetória, nos auxiliando em tomadas de decisões e acreditando no nosso potencial.*

# AGRADECIMENTOS

Nossos agradecimentos a todos os que contribuíram, direta ou indiretamente, para a realização deste trabalho, em especial:

Nosso orientador Prof. Jonas, que sempre esteve ao nosso orientando e dando conselhos para a realização do trabalho.

Aos professores Alex, Johanny, Pedro e Thiago, que durante toda essa jornada se comprometeram a compartilhar conosco todo o conhecimento que carregam consigo.

Gostaríamos de gostaria de agradecer a Deus, que durante toda nossa vida tem nos acompanhado e ajudado nos momentos mais difíceis.

Agradecemos aos nossos pais, Kelly e Alex (Júlia), Renato e Suely (Larissa), que sempre apoiaram nossas decisões e nos deram embasamento para chegar aonde chegamos.

Em particularidade de Júlia, agradeço ao meu namorado Gabriel que sempre me apoiou em todo momento.

Em particularidade de Larissa, agradeço ao meu marido Viniccius que sempre esteve ao meu lado me apoiando e me incentivando a nunca desistir. Agradeço também a minha filha Lorena que desde que nasceu vem sendo a minha maior motivação para continuar sempre em busca do crescimento tanto pessoal, quanto profissional. E sou grata também ao meu irmão Felipe que em momentos de dificuldade sempre me ajudou a encontrar uma solução.

O maior problema em informática não é o hardware ou o software, quase sempre está localizado entre a cadeira e o monitor (Hémerson Antônio da Silva)

# RESUMO

Este estudo tem como objetivo o desenvolvimento do projeto de uma câmera de segurança inteligente de baixo custo, tendo em vista a proteção contra invasões domiciliares através de sistemas de segurança inteligentes e acessíveis. Moradores de centros urbanos vivenciam a violência e a criminalidade nas ruas diariamente, mas uma ameaça em especial tem o poder de violar o seu local mais seguro e inviolável, a invasão de domicílio. Este tipo de crime não apenas pode causar danos materiais, mas também abalar o estado emocional das vítimas, fazendo com que acreditem que seu lar não é mais um lugar seguro. Para assegurar que isso não aconteça, várias medidas que visam manter a segurança e integridade das residências foram criadas e popularizadas, como a fortificação de fechaduras e instalação de portões elétricos controlados por dispositivos e de cercas eletrificadas. Entretendo, uma dessas medidas tem se tornado cada vez mais popular à medida que acompanha a evolução da tecnologia, os dispositivos de monitoramento inteligente. Através da captação e análise das imagens coletadas é possível identificar qualquer tipo de mudança em um ambiente, nesse caso, identificar um indivíduo estranho se infiltrando em um local particular. Contudo, a adoção dessa medida representa um alto custo para o usuário, tendo em vista o custo da tecnologia da câmera que irá captar as imagens e a complexidade da construção de um sistema inteligente para analisar as imagens. Nesse sentido, este trabalho propõe a criação de uma câmera de segurança inteligente e de baixo custo, adotando materiais mais baratos e disponibilizando um sistema de análise de imagens inteligente, que pode ser usado por quaisquer tipos de usuários.

**Palavras-chave:** Segurança, Monitoramento, Visão Computacional

# LISTA DE ILUSTRAÇÕES

[Figura 1 – Esp 32 Cam, Módulo Adaptador Mb e Câmera Ov2640, componentes separados 22](#_Toc169018374)

[Figura 2 – Esp 32 Cam, Módulo Adaptador Mb e Câmera Ov2640, componentes juntos 22](#_Toc169018375)

[Figura 3 – Cabo micro USB 23](#_Toc169018376)

[Figura 4 – Esp 32 Cam conectado a um computador 23](#_Toc169018377)

[Figura 5 – Esp 32 Cam conectado a uma fonte de carregador 24](#_Toc169018378)

[Figura 6 – Módulo Antena Transceptor ESP8266 MT7681 3DBI 25](#_Toc169018379)

[Figura 7 – Trecho de código que inicia a captura da imagem do módulo Esp32 Cam 26](#_Toc169018380)

[Figura 8 – Trecho de código que finaliza a captura da imagem do módulo Esp32 Cam 26](#_Toc169018381)

[Figura 9 – Trecho de código que converte a primeira imagem capturada para a escala preto e branco 27](#_Toc169018382)

[Figura 10 – Trecho de código que realiza a detecção de movimento nas imagens 27](#_Toc169018383)

[Figura 11 – Trecho de código que armazena a imagem capturada em uma pasta especifica com um nome único 27](#_Toc169018384)

[Figura 12 – Interface do software 28](#_Toc169018385)

[Figura 13 – pop-up da interface do software 29](#_Toc169018386)

[Figura 14 – Alteração no script para aumentar a claridade da imagem 31](#_Toc169018387)

[Figura 15 –script de instalação automática de bibliotecas sendo executado 31](#_Toc169018388)

[Figura 16 – Tela inicial da interface do software 32](#_Toc169018389)

[Figura 17 – Aviso pop-up de requisitos para iniciar o monitoramento 33](#_Toc169018390)

[Figura 18 – Aba de monitoramento 34](#_Toc169018391)

[Figura 19 – Aba de monitoramento ao detectar um movimento 34](#_Toc169018392)

[Figura 20 – Pasta com as imagens com movimento detectado 35](#_Toc169018393)

[Figura 21 – Envio de mensagem automática do movimento detectado pelo Web WhatsApp 35](#_Toc169018394)

[Figura 22 – Aba de configuração geral do software 36](#_Toc169018395)

[Figura 23 – Aba de configuração inicial do software 36](#_Toc169018396)

[Figura 24 – Aviso pop-up de requisitos para inserir o IP 37](#_Toc169018397)

[Figura 25 – pop-up para inserir o IP 38](#_Toc169018398)

[Figura 26 – Arquivo de texto que armazena o IP 38](#_Toc169018399)

[Figura 27 – Aviso pop-up de requisitos para inserir o número de telefone 39](#_Toc169018400)

[Figura 28 – pop-up para inserir o número de telefone 39](#_Toc169018401)

[Figura 29 – Arquivo de texto que armazena o número de telefone 40](#_Toc169018402)

[Figura 30 – Aviso pop-up com informações sobre a conexão 40](#_Toc169018403)

[Figura 31 – Página web do Web WhatsApp 41](#_Toc169018404)

[Figura 32 – Aba de testes do software 41](#_Toc169018405)

[Figura 33 – Aviso pop-up de requisitos para realizar o teste de imagem 42](#_Toc169018406)

[Figura 34 – Aba do teste de imagem 42](#_Toc169018407)

[Figura 35 – Aviso pop-up de requisitos para realizar o teste de monitoramento 43](#_Toc169018408)

[Figura 36 – Aba do teste de monitoramento 43](#_Toc169018409)

[Figura 37 – Aviso pop-up de requisitos para realizar o teste de envio de mensagens 44](#_Toc169018410)

[Figura 38 – Mensagem de teste enviada 44](#_Toc169018411)

[Figura 39 – Materiais da estrutura externa 45](#_Toc169018412)

[Figura 40 – Canos com furo para a câmera 45](#_Toc169018413)

[Figura 41 – Canos com furo antena e cabo micro USB 46](#_Toc169018414)

[Figura 42 – Canos com tinha spray na cor preta 46](#_Toc169018415)

[Figura 43 – Cano menor fixado no cano maior 47](#_Toc169018416)

[Figura 44 – Cano menor fixado no cano maior de frente 47](#_Toc169018417)

[Figura 45 – Cano menor com manta de polietileno expandido e módulo Esp32 Cam 48](#_Toc169018418)

[Figura 46 –Módulo Esp32 Cam alinhado com o furo dos canos 48](#_Toc169018419)

[Figura 47 – Antena conectada na parte interior 49](#_Toc169018420)

[Figura 48 – Antena conectada na parte exterior 49](#_Toc169018421)

[Figura 49 - Cabo micro USB conectado na parte interior 50](#_Toc169018422)

[Figura 50 - Cabo micro USB conectado na parte exterior 50](#_Toc169018423)

[Figura 51– Câmeras finalizada vista de frente 51](#_Toc169018424)

[Figura 52 – Câmeras finalizada vista de costas 51](#_Toc169018425)

# LISTA DE ABREVIATURAS E SIGLAS

WI-FI – Wireless Fidelity

RGB – Red Green Blue

HD – High Definition

IDE – Integrated Development Environment

LED – Light-Emiting Diode

SD – Secure Digital Card

USB – Universal Serial Bus

PVC – Polímero Policloreto de Vinila

BITS – Binary Digit

OpenCV – Open Computer Vision Library

SEJUSP – Secretaria de Estado de Justiça e Segurança Pública

IA – Inteligência Artificial

IP – Internet Protocol

# SUMÁRIO

[AGRADECIMENTOS 6](#_Toc169189586)

[RESUMO 8](#_Toc169189587)

[LISTA DE ILUSTRAÇÕES 9](#_Toc169189588)

[LISTA DE ABREVIATURAS E SIGLAS 11](#_Toc169189589)

[SUMÁRIO 12](#_Toc169189590)

[1 INTRODUÇÃO 14](#_Toc169189591)

[2 OBJETIVO 16](#_Toc169189592)

[3 REFERENCIAL TEÓRICO 16](#_Toc169189593)

[3.1 Sistemas de Segurança 16](#_Toc169189594)

[3.2 Visão Computacional 17](#_Toc169189595)

[4 TRABALHOS CORRELATOS 19](#_Toc169189596)

[4.1 Câmera Nest Cam 19](#_Toc169189597)

[4.2 Câmera Ring Indoor Cam 20](#_Toc169189598)

[5 MATERIAIS E MÉTODOS 21](#_Toc169189599)

[5.1 Módulo Esp32 Cam *Wi-Fi* Bluetooth com Câmera Ov2640 e Módulo Adaptador Mb 21](#_Toc169189600)

[5.2 Módulo Antena Transceptor ESP8266 MT7681 3DBI 24](#_Toc169189601)

[5.3 Python 25](#_Toc169189602)

[5.4 OpenCV 26](#_Toc169189603)

[5.5 Interface 28](#_Toc169189604)

[5.6 WhatsApp 29](#_Toc169189605)

[5.7 Estrutura externa 29](#_Toc169189606)

[6 RESULTADOS 30](#_Toc169189607)

[6.1 Programação e configuração do módulo Esp32 Cam 30](#_Toc169189608)

[6.2 Instalação das bibliotecas 31](#_Toc169189609)

[6.3 Interface 32](#_Toc169189610)

[6.3.1 Início 32](#_Toc169189611)

[6.3.2 Configuração Geral 35](#_Toc169189612)

[6.3.3 Configuração Inicial 36](#_Toc169189613)

[6.3.4 Testes 41](#_Toc169189614)

[6.4 Estrutura externa 45](#_Toc169189615)

[6.5 Funcionamento do projeto 51](#_Toc169189616)

[6.6 GitHub e Manual do usuário 52](#_Toc169189617)

[7 CONCLUSÃO 53](#_Toc169189618)

[7.1 Trabalhos futuros 54](#_Toc169189619)

[REFERÊNCIAS 55](#_Toc169189620)

[8 Bibliografia 55](#_Toc169189621)

[APÊNDICE 57](#_Toc169189622)

# INTRODUÇÃO

Nos últimos anos, temos testemunhado um notório aumento nos problemas relacionados à segurança. Diariamente, somos bombardeados por relatos de ataques e invasões domiciliares em veículos de comunicação, evidenciando uma realidade preocupante. Esta crescente incidência de crimes contra a segurança residencial e pessoal, levanta sérias preocupações e destaca a urgência de abordagens eficazes para proteger os lares e as pessoas que neles habitam.

Diante desse cenário crescente de criminalidade, torna-se cada vez mais necessário buscar recursos que fortaleçam a segurança das residências e de seus habitantes. Com o avanço notável da tecnologia, atualmente dispomos de meios eficazes para reforçar a proteção domiciliar, como sistemas de segurança e câmeras de vigilância. No entanto, um dos principais obstáculos enfrentados pela população é o alto custo desses dispositivos, imposto pelas empresas fornecedoras. Muitas vezes, os produtos disponíveis no mercado não correspondem plenamente às expectativas de segurança, o que gera uma discrepância entre o investimento realizado e os benefícios obtidos.

O propósito fundamental deste estudo é assegurar que todos tenham acesso aos meios necessários para se proteger de forma eficaz contra possíveis ataques, sem que isso represente um gasto significativo. É amplamente reconhecido que, à medida que o tempo avança, tanto a inflação quanto a carga tributária tendem a aumentar, ao passo que os salários das famílias brasileiras sofrem uma constante desvalorização. Nesse contexto, a questão da segurança não é o único desafio enfrentado pelas famílias tradicionais do Brasil. Portanto, o maior obstáculo reside em desenvolver soluções eficientes e acessíveis, permitindo que todos tenham meios de proteção sem comprometer suas finanças.

Segundo uma publicação do site G1 Globo, os dados fornecidos pela Secretaria de Estado de Justiça e Segurança Pública (Sejusp) apontam que de janeiro a novembro de 2022, foram contabilizadas 2.983 ocorrências de invasão a residências apenas no estado de Minas Gerais, já no ano de 2023 houve um aumento de 11,33% na incidência desse tipo de crime cometido, com base em análises feitas no mesmo período o número subiu para 3.321 casos registrados apenas no estado de Minas Gerais (Zuba & Paula Proença, 2023).

De acordo com dados do Anuário Brasileiro de Segurança Pública de 2022, o Brasil registrou mais de 200 mil casos de roubo a residências ao longo do ano anterior. Esses números representam um aumento significativo em comparação com anos anteriores e evidenciam a gravidade do problema da invasão domiciliar no país (Anuário Brasileiro de Segurança Pública , 2023).

Esses dados refletem não apenas a frequência das invasões domiciliares, mas também o impacto psicológico que esses eventos têm sobre os moradores. A sensação de vulnerabilidade e medo dentro de casa pode ter efeitos profundos na qualidade de vida e no bem-estar emocional das pessoas afetadas.

Diante desse cenário, medidas de prevenção e combate à invasão domiciliar são essenciais para garantir a segurança e tranquilidade dos cidadãos. Investimento em implementação de sistemas de segurança residencial pode contribuir para reduzir os índices de invasões domiciliares e promover um ambiente mais seguro para todos.

# OBJETIVO

Desenvolvimento de um projeto de câmera inteligente de baixo custo. Uma câmera que é capaz de enviar dados através da conexão sem fio Wi-Fi.

A câmera pode ser deixada no ambiente interno da residência desde que esse local tenha acesso à Internet. As imagens capturadas pela câmera deverão ser enviadas para um *software* em um computador, o *software* realizara a análise das imagens enviadas e deverá permanecer em execução para o perfeito funcionamento do monitoramento. Já o computador que executará o *software* deverá estar conectado à internet, seja por conexão a cabo ou sem fio *Wi-Fi*, na mesma rede em que a câmera estará conectada.

Toda vez que o *software* detectar um movimento presente no campo de visão câmera, ele irá salvar a última imagem analisada, momento em que ocorre a movimentação, e irá enviá-la como mensagem de imagem pelo aplicativo de mensagens WhatsApp, as mensagens serão enviadas para um número de telefone previamente inserido pelo usuário.

O *software* irá possuir uma interface gráfica para facilitar seu uso e configuração, nela estarão presentes os menus: INICIAR, TESTES e CONFIGURAÇÃO.

A câmera e o *software* preferencialmente deverão ser iniciados quando os residentes da casa não estiverem presentes na mesma, isso garantira o bom funcionamento do monitoramento da residência.

# REFERENCIAL TEÓRICO

## Sistemas de Segurança

Sistemas de segurança são um conjunto de *hardwares* e *softwares* que atuam em conjunto para monitorar e prevenir invasões ou furtos nos locais em que são instalados. Atualmente existem diversos tipos de sistemas de segurança no mercado, tanto para pequenos espaços como residências, quanto para espaços em larga escala como empresas.

Os sistemas de segurança empresarial têm como objetivo proteger os ativos e as operações internas da empresa de ameaças como acessos não autorizados a áreas restritas e furtos de informação ou matérias, geralmente esses sistemas são compostos de um conjunto de câmeras e detectores para monitorar e analisar os ambientes internos e externos da empresa.

Já os sistemas de segurança residências atuam para prevenir que a casa seja invadida ou que tenha os bens roubados, eles geralmente contam com uma ou mais câmeras que monitoram a parte externa ou interna da residência. Os modelos mais atuais de câmeras para segurança residencial possuem conexão sem fio *Wi-Fi*, detectores internos de movimento, visão noturna, caixa de som, luzes *led* e podem ser remotamente rotacionadas, o que aumenta seu campo de visão, além de possuírem aplicativos móveis para que os usuários possam ver as imagens da câmera em tempo real e falar através da caixa de som.

Entretanto, quanto mais tecnologias são atribuídas a essas câmeras, maior o seu custo final para o usuário, devido ao valor e a complexidade dessas tecnologias.

## Visão Computacional

A visão é um dos sentidos mais utilizados no dia a dia, seja ao atravessar uma rua, ler um livro ou para dirigir. Até então toda atividade realizada através da visão poderia ser feita apenas pelo olho humano, até o surgimento das câmeras fotográficas. Pela primeira vez era possível capturar imagens do mundo real e segurá-las na palma da mão, logo após sugiram as câmeras filmadoras que capturavam várias imagens por segundo e as executavam em sequência rapidamente para dar o efeito de movimento e assim criando um vídeo. Anos depois foram criadas câmeras filmadoras que tinham capacidade de receber imagens e reproduzia-las sequencialmente em tempo real, ou seja, um vídeo em tempo real. Uma das áreas que mais se beneficiaram com essa tecnologia foi a área da segurança, graças ao vídeo em tempo real era possível que uma única pessoa vigiasse várias áreas diferentes ao mesmo tempo e em ângulos que seriam impossíveis para uma pessoa enxergar sozinha, já que as câmeras poderiam ser instaladas em diversas superfícies, como lugares altos e inacessíveis.

Entretendo, errar é humano e muitas vezes coisas podem passar desapercebidas nas filmagens ou demoramos para compreender algo que foi capturado pela câmera, foi a partir dessas e de outras necessidades que a visão computacional surgiu.

A visão computacional é uma área da inteligência artificial que estuda e desenvolve máquinas que podem capturar e analisar imagens de forma automatizada e inteligente.

O autor Ricardo Antonello cita em seu livro “Introdução a Visão Computacional com Python e OpenCV”, que “a visão computacional é a ciência que aborda a tecnologia das máquinas que enxergam, ela é focada no desenvolvimento de sistemas que funcionam recebendo dados através de imagens ou de qualquer dado multidimensional” (Antonello, 2017).

Alguns dos conceitos e técnicas abordados incluem:

Detecção de Objetos: Sistemas de detecção de objetos que coletam dados e os processam em um algoritmo de detecção de objetos para identificar pessoas, animais ou objetos específicos, essa detecção pode ser feita através de redes neurais artificiais, com é o caso do Ultralytics YOLO. Primeiramente é necessário realizar a classificação das imagens para associar os objetos a uma classe e depois localizar esses objetos nas imagens coletadas, geralmente os objetos são circulados com um contorno em uma cor vibrante.

Reconhecimento Facial: O reconhecimento facial é uma técnica de biometria baseada na forma do rosto humano, diferente da detecção de objetos, essa técnica é focada em classificar e localizar rostos humanos a partir de uma ligação geométrica de traços e tamanhos. Alguns dos algoritmos de reconhecimento facial usam do aprendizado de máquina, área da IA focada na capacidade de máquinas de adquirir conhecimento, que consiste em apresentar várias imagens de rostos humanos e fazer com que a máquina reconheça padrões nessas imagens, para que assim ela possa reconhecer e localizar esses padrões em qualquer imagem.

Detecção de Movimento: A detecção de movimento é uma técnica que aborda o monitoramento de imagens a fim de identificar e localizar mudanças no ambiente. Quando uma imagem for capturada, ela é comparada com a imagem seguinte, nessa comparação ocorre a subtração dos *pixels* da imagem e caso exista diferença, um movimento é detectado. Quando um movimento for detectado, o algoritmo localiza os pixels em que o movimento ocorreu e traça um contorno em volta deles, geralmente usando cores vibrantes.

Existe ainda uma necessidade de se realizar o tratamento das imagens antes que elas sejam processadas pelos algoritmos de visão computacional, para facilitar sua análise e evitar erros de detecção e classificação de imagens, uma técnica muito adotada é de aplicar filtros de cores que convertem imagens coloridas em imagens na escala de cores preto e branco. Isso é feito porque uma imagem na escala preto e branco tem apenas um canal (uma matriz de 2 dimensões), enquanto imagens coloridas, padrão RGB, tem 3 canais (uma matriz de 3 dimensões), ou seja, ao converter a imagem para a escala preto e branco estamos simplificando a imagem para que seja mais facilmente processada, essa técnica também uniformiza falhas causadas pela iluminação da imagem.

# TRABALHOS CORRELATOS

Após realizar diversas buscas de produtos relacionados a câmeras de monitoramento inteligente, foi possível perceber que atualmente existe uma grande diversidade de opções que podem atender a diversos tipos de consumidores, dependendo da necessidade de cada um. Abaixo está detalhado alguns dos produtos existentes no mercado.

## Câmera Nest Cam

 Foi projetada pela empresa Nest, que é uma subsidiária da Google e oferece os seguintes benefícios:

Qualidade de Vídeo em Alta Definição: A Nest Cam oferece vídeo em alta definição (HD) para capturar imagens claras e nítidas. Isso permite aos usuários monitorarem sua residência com detalhes precisos.

Detecção de Movimento e Notificações: Equipada com tecnologia de detecção de movimento, a Nest Cam pode alertar os usuários sempre que detectar atividade em sua área de monitoramento. Essas notificações em tempo real permitem que os usuários estejam cientes de eventos importantes em sua casa, mesmo quando estão longe.

Visão Noturna: A Nest Cam possui visão noturna, permitindo a captura de imagens mesmo em condições de pouca luz ou escuridão total. Isso garante que a segurança da residência não seja comprometida durante a noite.

Integração com Ecossistema Doméstico Inteligente: Como parte do ecossistema doméstico inteligente da Nest, a Nest Cam pode ser integrada a outros dispositivos inteligentes da Nest, como termostatos, detectores de fumaça e dispositivos de segurança adicionais. Isso permite uma automação mais abrangente e controle unificado da casa.

Armazenamento em Nuvem: A Nest Cam oferece opções de armazenamento em nuvem para gravar e acessar vídeos gravados. Isso permite aos usuários revisarem o histórico de gravações e eventos anteriores, garantindo uma vigilância contínua da residência.

Assistência por Voz: A Nest Cam pode ser controlada e gerenciada por meio de comandos de voz quando integrada a assistentes de voz como o Google Assistant, proporcionando uma experiência mais conveniente e hands-free.

Esse modelo possui um valor de custo-benefício de aproximadamente R$1.500,00, sendo essa uma desvantagem, pois não é tão acessível a todos os públicos. Outra desvantagem são os custos contínuos que são necessários para o bom funcionamento dela, como por exemplo o valor de serviço de armazenamento em nuvem.

## Câmera Ring Indoor Cam

Foi projetada pela empresa Ring e oferece os seguintes benefícios.

Resolução de vídeo: A Ring Indoor Cam geralmente grava em resolução HD, oferecendo uma qualidade de imagem nítida para monitoramento interno.

Campo de visão: Possui um amplo campo de visão, permitindo que você monitore uma área maior dentro de sua casa.

Conectividade: A câmera normalmente se conecta à rede *Wi-Fi* de sua casa, permitindo o acesso remoto aos vídeos por meio do aplicativo Ring em smartphones ou tablets.

Áudio bidirecional: A maioria dos modelos possui áudio bidirecional, o que significa que você pode ouvir o que está acontecendo na área monitorada e falar através da câmera usando o aplicativo.

Visão noturna: Geralmente equipada com visão noturna infravermelha, permitindo que você monitore sua casa mesmo em condições de pouca luz ou escuridão completa.

Detecção de movimento: A Ring Indoor Cam normalmente possui recursos de detecção de movimento, enviando notificações para o seu dispositivo móvel quando ativada por movimento dentro do campo de visão da câmera.

Armazenamento de vídeo: Os vídeos gravados geralmente podem ser armazenados na nuvem com uma assinatura do serviço Ring Protect, oferecendo a capacidade de revisitar e revisar vídeos passados.

Compatibilidade com assistentes de voz: Muitas vezes, a Ring Indoor Cam é compatível com assistentes de voz como Amazon Alexa, permitindo que você controle a câmera usando comandos de voz.

Esse modelo possui um valor de custo-benefício de aproximadamente R$800,00, sendo essa uma desvantagem, pois não é tão acessível a todos os públicos. Assim como o produto anterior, outra desvantagem são os custos contínuos que são necessários para uma melhor performance , como por exemplo o valor de serviço de armazenamento em nuvem. Além disso, para esse produto, caso o usuário tenha o desejo de obter recursos mais avançados, como detecção personalizada de movimentos ou acesso aos vídeos gravados por um período de tempo maior, ele precisará pagar um valor mensal/anual de assinatura para se utilizar deles.

# MATERIAIS E MÉTODOS

## Módulo Esp32 Cam *Wi-Fi* Bluetooth com Câmera Ov2640 e Módulo Adaptador Mb

O Módulo Esp32 Cam *Wi-Fi* Bluetooth é uma versão do Esp32, um *chip* microcontrolador dual-core de 32 bits com *Wi-Fi* e *Bluetooth*, que é adaptado especificamente para a instalação de uma Câmera Ov2640 ou OV7670.

Esse módulo conta com *Wi-Fi* e Bluetooth com antena integrada, suporte para cartão SD e para as linguagens de programação LUA e C++ na IDE do Arduino, luz flash em LED e 4 MB de Memória Flash programável, 11 pinos digitais, resolução de foto de 2 Megapixels e um *slot* para inserir a câmera.

Para que o módulo seja programado, ele precisa de um adaptador com conexão para cabo micro USB, para que a entrada USB seja conectada a um computador que possui o *script* de programação que será enviado para o módulo. As versões mais recentes do módulo Esp32 Cam são vendidas junto de um módulo Adaptador Mb, ele se conecta nos pinos na parte inferior do Esp32 Cam e possui uma entrada micro USB e dois botões: IO0 e RST, sendo o RST um recurso essencial para reiniciar o módulo após o *upload* do *script*.

O módulo Esp32 Cam foi escolhido para o projeto pois é perfeito para protótipos de projetos que abordam a visão computacional, possui um baixo custo para a tecnologia que possui e pode se conectar remotamente através do sinal *Wi-Fi*.

Figura 1 – Esp 32 Cam, Módulo Adaptador Mb e Câmera Ov2640, componentes separados

Tela de um aparelho eletrônico

Descrição gerada automaticamente com confiança baixa

Fonte: Bitmaker (2024)

Figura 2 – Esp 32 Cam, Módulo Adaptador Mb e Câmera Ov2640, componentes juntos

Circuito eletrônico com fios

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Para que seja possível inserir *script*s de programação no Módulo Esp32 Cam, um cabo micro USB será conectado no Esp32 Cam, com a ponta micro USB do cabo, e a ponta USB do cabo será conectada ao computador que possui o *script*.

Figura 3 – Cabo micro USB

Mouse de computador

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

Figura 4 – Esp 32 Cam conectado a um computador



Fonte: Elaboração do próprio autor

O cabo micro USB também será usado para alimentar o Esp32 Cam, para isso a ponta micro USB do cabo deverá ser conectada no Esp32 Cam e a ponta USB do cabo deverá ser conectada a uma fonte de carregador que será colocada em uma tomada.

Figura 5 – Esp 32 Cam conectado a uma fonte de carregador

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

O cabo micro USB e a fonte de carregador utilizados no projeto vieram de um carregador de celular comum facilmente encontrada.

## Módulo Antena Transceptor ESP8266 MT7681 3DBI

A Antena Transceptor ESP8266 MT7681 3DBI é uma antena de sinal *Wi-Fi* com uma frequência de 2,4 GHz e possui suporte a uma variedade de placas microcontroladoras, como: Arduino, Raspberry Pi e Esp32 Cam.

Ela foi escolhida para o projeto para amplificar o sinal do módulo Esp32 Cam, além de garantir que o sinal não seja afetado quando o Esp32 Cam estiver dentro da sua estrutura de proteção externa, já que a antera ficara na parte de fora da estrutura.

Figura 6 – Módulo Antena Transceptor ESP8266 MT7681 3DBI

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## Python

Python é uma linguagem de alto nível orientada a objetos, interpretada por *script*s e com uma tipagem forte. Essa linguagem de programação foi escolhida para ser utilizada no projeto por ser uma linguagem amplamente usada, de fácil aprendizado e por possuir uma sintaxe enxuta, o que contribui para melhorar sua performance.

Essa linguagem também possui uma diversidade de bibliotecas instaláveis que ampliam ainda mais as suas funcionalidades. As bibliotecas usadas nesse projeto são:

1. Opencv-python – Biblioteca que realiza captação e exibição de imagens enviadas pelo Esp32 Cam, além de realizar o monitoramento de detecção de movimento.

2. Pywhatkit – Biblioteca que realiza o envio das mensagens e imagens via WhatsApp.

3. Pywin32 – Biblioteca necessária para garantir o bom funcionamento da biblioteca Pywhatkit.

4. Customtkinter – Biblioteca para criação de interfaces, tem como base a biblioteca Tkinter.

5. CTkMessagebox – Biblioteca para criação de caixas de texto *pop-ups* para interfaces criadas com a biblioteca Customtkinter.

6. Time – Biblioteca que pode recuperar a hora e data atual.

7. Os – Biblioteca que pode realizar atividades no sistema operacional, como salvar arquivos, acessar arquivos e pastas, criar e deletar pastas.

8. PIL – Biblioteca usada para importar arquivos de imagem para interface criada na biblioteca Customtkinter.

9. Webbrowser – biblioteca usada para abrir endereços web.

10. Subprocess – biblioteca usada para baixar automaticamente todas as bibliotecas necessárias para o projeto em conjunto com a biblioteca Sys.

11. Sys – biblioteca usada para baixar automaticamente todas as bibliotecas necessárias para o projeto em conjunto com a biblioteca Subprocess.

## OpenCV

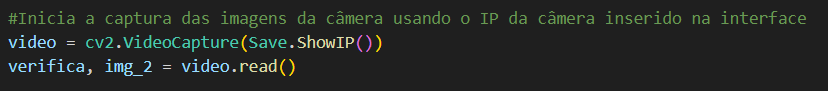
O OpenCV é uma biblioteca de código aberto focada na área da visão computacional, aprendizado de máquina e processamento de imagens. Originalmente criada pela empresa Intel, atualmente ela é mantida pela comunidade de desenvolvedores da OpenCV Foundation e tem o uso gratuito tanto para a área comercial quanto a área acadêmica. Essa biblioteca é escrita nativamente na linguagem de programação C++, mas possui suporte para Java, MATLAB e Python.

Essa biblioteca foi escolhida para o projeto, pois além de sem uma biblioteca conhecida por ser utilizada amplamente em projetos de visão computacional, ela também possui uma variedade de funções que facilitam muito a construção de um algoritmo de visão computacional.

Nesse projeto a biblioteca foi usada para o desenvolvimento de três funcionalidades da câmera de segurança:

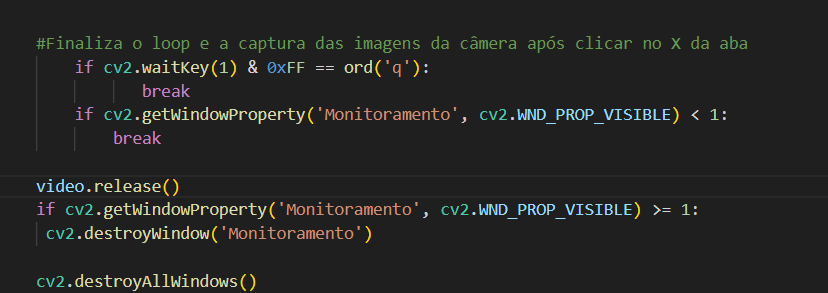
1. Iniciar e finalizar a captura das imagens da câmera

Figura 7 – Trecho de código que inicia a captura da imagem do módulo Esp32 Cam



Fonte: Elaboração do próprio autor

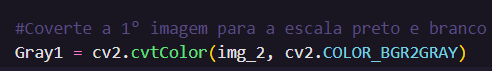
Figura 8 – Trecho de código que finaliza a captura da imagem do módulo Esp32 Cam



Fonte: Elaboração do próprio autor

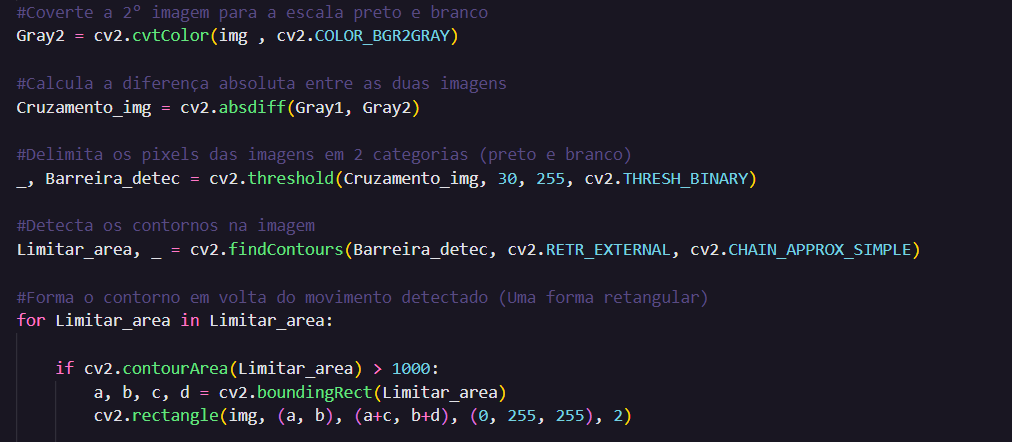
1. Realizar a detecção de movimento

Figura 9 – Trecho de código que converte a primeira imagem capturada para a escala preto e branco



Fonte: Elaboração do próprio autor

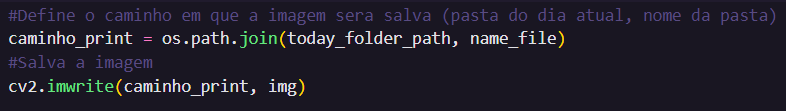
Figura 10 – Trecho de código que realiza a detecção de movimento nas imagens



Fonte: Elaboração do próprio autor

1. Salvar imagens da câmera com um nome específico em uma pasta escolhida

Figura 11 – Trecho de código que armazena a imagem capturada em uma pasta especifica com um nome único



Fonte: Elaboração do próprio autor

## Interface

A interface do *software* foi feita com a biblioteca Customtkinter, que foi construída com base na biblioteca Tkinter, biblioteca popularmente usada para interfaces na linguagem de programação Python.

A Biblioteca Customtkinter foi escolhida para o projeto, pois ela possui uma vasta variedade de elementos visuais e funcionalidades que se alinhavam com os objetivos esperados da interface final do projeto.

Foi usada também a biblioteca CTkMessagebox que foi usada para a criação de mensagens de texto *pop-ups* que exibissem informações e recebessem entradas de texto.

Figura 12 – Interface do software

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 13 – pop-up da interface do software

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## WhatsApp

O WhatsApp é um aplicativo multiplataforma de mensagens digitais instantâneas e ligações por voz e vídeo, além de enviar mensagens de texto ele também pode enviar imagens, vídeos, documentos e arquivos de áudio. Para que possa ser utilizado, é necessário instalar o aplicativo WhatsApp Messenger em um dispositivo móvel e possuir um número de telefone atrelado ao aparelho. Já para utilizá-lo em um computador é preciso entrar na página *web* do Web WhatsApp em um navegador e entrar a sua conta do WhatsApp Messenger através de um código QR.

O WhatsApp foi escolhido por sua popularidade como um aplicativo de mensagens instantâneas, além de eliminar a necessidade do usuário de precisar instalar e se familiarizar com um aplicativo próprio para utilizar a câmera. Uma vez que ele poderia utilizar o WhatsApp Messenger, que é um aplicativo que já estaria instalado em seu dispositivo móvel e que ele já possuiria familiaridade.

## Estrutura externa

A construção de uma estrutura externa para a câmera é essencial para garantir a sua proteção, por mais que o propósito da câmera seja funcionar em um ambiente interno, é fundamental garantir a sua proteção contra umidade, calor e quaisquer outros fatores que possam danificar a câmera ou prejudicar seu funcionamento.

O material escolhido para a construção dessa estrutura externa foi o PVC, uma vez que ele é um material resistente, moldável e pode ser facilmente achado em qualquer loja de construção.

A estrutura contara com um cano PVC grande com 7,5 11 centímetros de diâmetro e 11 centímetros de altura, um cano PVC pequeno com 5 centímetros de diâmetro e 5,5 centímetros de altura, duas tampas para cano PVC com 8 centímetros de diâmetro e 4,5 centímetros de altura, um pedaço de manta de polietileno expandido e cola própria para cano PVC.

A manta de polietileno expandido é um material plástico leve e rígido, ele foi escolhido para o projeto para proteger e manter a câmera dentro da estrutura externa, já que ele possui propriedades para proporcionar proteção contra riscos e controle de temperatura. Ele pode ser facilmente achado em embalagens, já que é um material indicado para o embalo de peças industriais.

# RESULTADOS

O projeto desenvolvido foi o de um módulo Esp32 Cam que captura imagens em formato de vídeo em tempo real e realiza uma transmissão que pode ser acessada através de um endereço IP, um *software* desenvolvido na linguagem Pyhton recebe essas imagens e realiza um monitoramento de detecção de movimento em tempo real, caso o *software* identifique um movimento no ambiente ele irá salvar localmente as imagens em que a movimentação ocorre e irá enviá-las para um número de telefone previamente informado via WhatsApp Web.

## Programação e configuração do módulo Esp32 Cam

A programação do módulo Esp32 Cam foi feita através do *software* IDE Arduino com a extensão “esp32 by Espressif Systems’’. O *script* usado foi o “CameraWebServer”, disponível no próprio programa dentro da aba Exemplos na aba *File*, ele permite que o Esp32 Cam tenha um IP próprio e que suas imagens possam ser vistas e configuradas por meio de uma página *web* em um navegador. Uma alteração foi feita no *script* original para aumentar a claridade das imagens, sendo ela:

Figura 14 – Alteração no script para aumentar a claridade da imagem



Fonte: Elaboração do próprio autor

Após realizar o *upload* do arquivo no Esp32 Cam, o painel serial irá exibir um IP específico para ele. Ao inserir esse IP em um navegador, ele abre uma página *web* onde algumas configurações sobre a transmissão podem ser alteradas, como a qualidade da imagem exibida e filtros que podem ser adicionados à imagem. Perto do fim da página existe o botão “Start Stream” que ao ser pressionado exibe as imagens capturadas pelo Esp32 Cam em tempo real com as configurações aplicadas. Para usar as imagens da Esp32 Cam no projeto foi necessário achar um meio de se obter as imagens da transmissão automaticamente, sem que fosse preciso apertar o botão “Start Stream”. Para isso a terminação “:81/stream” foi adicionada no endereço de IP, formando “123.123.1.123:81/stream”

## Instalação das bibliotecas

Para facilitar a instalação das bibliotecas necessárias para executar o *software*, um *script* foi desenvolvido, basta executá-lo e ele irá instalar todas a bibliotecas Python necessárias automaticamente no computador. Quando terminar, ele irá fechar automaticamente.

Figura 15 –script de instalação automática de bibliotecas sendo executado

Texto

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## Interface

## 6.3.1 Início

A aba início possui um botão que inicia o monitoramento, exibe o número de telefone e o IP do módulo Esp32 Cam inseridos pelo usuário e uma ilustração feita para o projeto.

Figura 16 – Tela inicial da interface do software

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Quando o Botão Iniciar Monitoramento é clicado, uma mensagem de texto *pop-up* é exibida contendo alguns requisitos que devem ser atendidos para garantir um bom funcionamento do monitoramento.

Figura 17 – Aviso pop-up de requisitos para iniciar o monitoramento

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

O *script* do monitoramento funciona da seguinte forma: O monitoramento é iniciado, um movimento é localizado e contornado em uma imagem, um nome único é dado a essa imagem, a data e hora da detecção junto com um número único e a terminação “\_Image.png”, ela é armazenada em uma pasta previamente criada e que foi nomeada com a data da execução do *script*, a página *web* WhatsApp Web é aberta na conta logada previamente pelo usuário, a imagem armazenada é enviada para o número de telefone previamente inserido pelo usuário e a página *web* é encerrada automaticamente, após o *script* enviar a sequência de imagens que juntas formavam o trecho de vídeo do movimento detectado, o monitoramento é encerrado, logo após o monitoramento é reiniciado e ciclo se repete até que o monitoramento seja permanentemente encerrado com um clique no botão “X” na aba.

Figura 18 – Aba de monitoramento

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 19 – Aba de monitoramento ao detectar um movimento

Interface gráfica do usuário

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 20 – Pasta com as imagens com movimento detectado

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança baixa

Fonte: Elaboração do próprio autor

Figura 21 – Envio de mensagem automática do movimento detectado pelo Web WhatsApp

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

## 6.3.2 Configuração Geral

A aba de configuração possui outras duas abas internas: as abas de configuração geral e configuração inicial. A aba de configuração geral possui o botão “Abrir pasta de imagens” que abre a pasta em que estarão localizadas as imagens com detecção de movimento, cada uma em uma pasta interna com a data da execução do programa, e o botão “Excluir todas as imagens” que excluirá todas as pastas com imagens armazenadas e abrira a pasta, agora vazia, em seguida.

Figura 22 – Aba de configuração geral do software

Tela de celular com fundo preto

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## 6.3.3 Configuração Inicial

Figura 23 – Aba de configuração inicial do software

Tela de um aparelho eletrônico

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

A aba de configuração inicial possui o botão “IP da Câmera” que abre uma mensagem de texto *pop-up* que informa alguns requisitos que devem ser atendidos para que possa ser possível inserir o IP do módulo Esp32 Cam. Após clicar no botão “Prosseguir”, uma nova aba *pop-up* será aberta para que o IP do Esp32 Cam seja inserido.

Ao ser inserido, ele será armazenado em um bloco de notas, para que possa ser guardado permanentemente ou até que um novo IP seja inserido nessa opção. Além ser armazenado, o IP também receberá os trechos de texto da seguinte forma: "http://"+Numero de IP+":81/stream", isso garante que ele funcione perfeitamente nos demais *script*s e poupa o usuário de ter que escrevê-los toda vez que adicionar um novo IP.

Para que o IP seja acessado futuramente, pelo *script* de monitoramento por exemplo, uma função foi criada para recuperar o número de IP da câmera e essa função retornara esse número toda vez que for chamada.

Figura 24 – Aviso pop-up de requisitos para inserir o IP

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 25 – pop-up para inserir o IP

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 26 – Arquivo de texto que armazena o IP

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Essa aba também possui um botão de “Registrar/Alterar número de telefone”, ele funciona de forma idêntica ao botão de “IP da Câmera”, mas ele recebe e armazena o número de telefone que será usado para o envio de mensagens e imagens via Web WhatsApp, ele também é formatado antes que seja armazenado da seguinte forma: “"+"+Número de telefone”

Figura 27 – Aviso pop-up de requisitos para inserir o número de telefone

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 28 – pop-up para inserir o número de telefone

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 29 – Arquivo de texto que armazena o número de telefone

Interface gráfica do usuário, Texto, Aplicativo, Word

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Por fim, essa aba possui um botão de “Conectar WhastApp” que ao ser clicado abre uma mensagem de texto *pop-up* que exibe algumas informações sobre como realizar a conexão com o WhatsApp Web. Após clicar no botão “Prosseguir”, a página *web* do WhatsApp Web é automaticamente aberta no navegador padrão do computador, caso o usuário já esteja com o WhatsApp Web logado no navegador, ele deve fechar a aba e voltar ao programa, caso ele não esteja logado, ele deve escanear o código QR da página *web* com o aplicativo WhatsApp Messenger.

Figura 30 – Aviso pop-up com informações sobre a conexão

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 31 – Página web do Web WhatsApp

Código QR

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## 6.3.4 Testes

Aba de testes possui alguns testes que podem ser realizados para verificar se a conexão do módulo Esp32 Cam, o *script* de monitoramento e o envio de mensagens via abre WhatsApp Web estão funcionando corretamente.

Figura 32 – Aba de testes do software

Tela de celular com fundo preto

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

A aba de testes possui o botão “Imagem” que abre uma mensagem de texto *pop-up* que informa alguns requisitos que devem ser atendidos para que possa ser possível realizar o teste de imagem da câmera.

Esse teste exibe a imagem capturada da câmera em tempo real e ele é encerrado quando o botão “X” da aba é clicado.

Figura 33 – Aviso pop-up de requisitos para realizar o teste de imagem

Tela de celular com fundo preto

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 34 – Aba do teste de imagem

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Seguidamente, essa aba possui o botão “Monitoramento” que abre uma mensagem de texto *pop-up* que informa alguns requisitos que devem ser atendidos para que possa ser possível realizar o teste de monitoramento da câmera.

Esse teste exibe a imagem capturada da câmera em tempo real e realiza a detecção de movimento nas imagens, quando um movimento é detectado, ele será contornado com uma forma retangular amarela. O teste é encerrado quando o botão “X” da aba é clicado.

Figura 35 – Aviso pop-up de requisitos para realizar o teste de monitoramento

Tela de celular

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 36 – Aba do teste de monitoramento

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Por fim, essa aba possui o botão “Envio de mensagens” que abre uma mensagem de texto *pop-up* que informa alguns requisitos que devem ser atendidos para que possa ser possível realizar o teste de envio de mensagens.

Esse teste abre automaticamente a página *web* WhatsApp Web em um navegador e envia uma mensagem de teste para o número de telefone previamente inserido pelo usuário, após enviar a mensagem, a página *web* é automaticamente encerrada.

Figura 37 – Aviso pop-up de requisitos para realizar o teste de envio de mensagens

Tela de celular com fundo preto

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 38 – Mensagem de teste enviada

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## Estrutura externa

A estrutura externa foi feita com um cano PVC grande com 7,5 11 centímetros de diâmetro e 11 centímetros de altura, um cano PVC pequeno com 5 centímetros de diâmetro e 5,5 centímetros de altura, duas tampas para cano PVC com 8 centímetros de diâmetro e 4,5 centímetros de altura, um pedaço de manta de polietileno expandido e cola própria pra PVC.

Figura 39 – Materiais da estrutura externa

Diagrama

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

Será feito um furo no cano maior e no cano menor para permitir que a câmera tenha acesso ao ambiente exterior para capturar imagens.

Figura 40 – Canos com furo para a câmera

Uma imagem contendo no interior, espelho, pequeno, mesa

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Serão feitos outros dois furos no cano maior para permitir a saída do cabo micro USB que alimentará a câmera e para permitir que a antena fique fora da estrutura, assim evitando interferências no sinal.

Figura 41 – Canos com furo antena e cabo micro USB

Uma imagem contendo no interior, pequeno, espelho, mesa

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Acabamento opcional: As partes externas dos dois canos de PVC e das tampas foram pintadas de preto por motivos estéticos, o material utilizado foi uma lata de tinta spray preta.

Figura 42 – Canos com tinha spray na cor preta

Mouse de computador

Descrição gerada automaticamente com confiança baixa

Fonte: Elaboração do próprio autor

O cano menor será inserido dentro do cano maior na altura do furo para alinhar os dois furos dos canos e será permanentemente colado com cola própria para PVC.

Figura 43 – Cano menor fixado no cano maior

Em preto e branco

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

Figura 44 – Cano menor fixado no cano maior de frente

Copo de café

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

O pedaço de manta de polietileno expandido será colocado dentro do cano pequeno junto com o módulo Esp32 Cam, é importante assegurar que a câmera do módulo esteja alinhada com os furos feitos nos dois canos.

Figura 45 – Cano menor com manta de polietileno expandido e módulo Esp32 Cam

Foto em preto e branco

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

Figura 46 –Módulo Esp32 Cam alinhado com o furo dos canos

Uma imagem contendo no interior, xícara, computador, café

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

A antena deve ser conectada ao módulo Esp32 Cam e deve ser montada do lado de fora do cano grande.

Figura 47 – Antena conectada na parte interior

Uma imagem contendo no interior, mesa, pequeno, prato

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 48 – Antena conectada na parte exterior

Copo de café

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

A ponta do cabo micro USB deve ser conectada ao módulo Esp32 Cam e o fio deve ser passado pelo furo feito no cano grande, de modo de que a saída USB fique no lado de fora.

Figura 49 - Cabo micro USB conectado na parte interior

Mouse de computador

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

Figura 50 - Cabo micro USB conectado na parte exterior

Mouse em cima da mesa

Descrição gerada automaticamente com confiança média

Fonte: Elaboração do próprio autor

As duas tampas PVC devem ser colocadas para fechar a estrutura interna.

Figura 51– Câmeras finalizada vista de frente

Uma imagem contendo no interior, mesa, pequeno, cozinha

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

Figura 52 – Câmeras finalizada vista de costas

Uma imagem contendo no interior, mesa, computador, pequeno

Descrição gerada automaticamente

Fonte: Elaboração do próprio autor

## Funcionamento do projeto

Primeiramente, o módulo Esp32 Cam deverá ser ligado, conectando a fonte de carregador em uma tomada, e a estrutura externa deverá ser colocada no local em que o usuário deseja monitorar. Seguidamente, o usuário deverá executar o *software* e realizar quaisquer testes ou configurações que quiser. Alguns testes e funcionalidades dependem de ações previamente realizadas pelo usuário, como inserção do IP do Esp32 Cam, inserção do número de telefone e *login* no WhatsApp Web. É de extrema importância que o usuário se atente a esses requisitos para o perfeito funcionamento do *software*.

Caso o usuário queira iniciar o monitoramento, ele deve clicar no botão “Iniciar Monitoramento” presente na aba Iniciar. Nesse momento o monitoramento de detecção de movimento irá começar e qualquer movimento detectado será capturado, armazenado e enviado para o usuário via WhatsApp Web. O usuário, se quiser, poderá deixar o monitoramento funcionando enquanto não está presente na residência, mas devera assegurar que o computador, que está executando o *software*, e a internet, que está conectando o Esp32 Cam e o WhatsApp Web, estejam ligados durante a sua ausência.

Quando o usuário quiser encerrar as atividades do Esp32 Cam e do monitoramento, ele deverá clicar no botão “X” na aba que estará exibindo a imagem do Esp32 Cam e devera clicar nesse mesmo botão na aba da interface do *software*. Após isso ele deve desconectar a fonte de carregador, que está alimentando a Eps32 Cam, da tomada.

## GitHub e Manual do usuário

Um dos principais objetivos desse projeto é o de garantir que ele possa ser replicado por qualquer pessoa, tenha ela familiaridade com a tecnologia ou não, para que este objetivo possa ser atingido, se faz necessário que os códigos e os arquivos do projeto sejam disponibilizados publicamente para que qualquer pessoa possa ter acesso a eles.

Para isso a plataforma de hospedagem de código e projetos GitHub foi escolhida para este projeto. Os códigos e os arquivos do projeto estão disponíveis publicamente no link do repositório abaixo:

<https://github.com/JuliaResendeF/GATUNO_Camera_de_Seguranca_Inteligente_de_Baixo_Custo.git>

Juntamente a isso, foi desenvolvido um manual do usuário para facilitar a reprodução do projeto e a familiarização do usuário com o funcionamento do *software*.

O manual estará disponível como documento no apêndice deste estudo.

# CONCLUSÃO

Ao longo deste trabalho, abordamos a crescente preocupação com a segurança residencial em vista do aumento alarmante de invasões domiciliares no Brasil. Os dados apresentados destacam a urgência de soluções acessíveis e eficazes para proteger os lares e garantir a tranquilidade das famílias. Com o avanço da tecnologia, a implementação de sistemas de segurança tem se mostrado uma medida indispensável, mas muitas vezes inacessível devido ao alto custo.

Neste contexto, o desenvolvimento de uma câmera inteligente de baixo custo surge como uma solução viável e inovadora. O projeto proposto de uma câmera capaz de enviar dados através de conexão *Wi-Fi* e equipada com um *software* de monitoramento, visa facilitar o acesso à segurança residencial. A capacidade de detectar movimentos e enviar alertas instantâneos via WhatsApp garante que os moradores possam reagir rapidamente a qualquer ameaça, mesmo à distância.

Os sistemas de segurança atuais, como exemplificado pelos produtos Nest Cam e Ring Indoor Cam, oferecem uma gama de funcionalidades avançadas, mas a um custo pode se tornar um obstáculo para muitas famílias. A proposta deste projeto é criar uma alternativa mais econômica, sem comprometer a eficiência e a qualidade da vigilância. Ao utilizar tecnologias de visão computacional, a câmera inteligente será capaz de identificar movimentos no ambiente, proporcionando um nível de segurança comparável aos sistemas de maior custo.

Além de atender à necessidade imediata de segurança, a câmera proposta também considera a facilidade de uso e a acessibilidade financeira. A interface gráfica intuitiva e a capacidade de configuração simplificada garantem que mesmo usuários com pouca experiência técnica possam utilizar o sistema de forma eficaz.

Este trabalho contribui para a redução da vulnerabilidade das residências brasileiras, oferecendo uma solução prática e acessível. Espera-se que, com a implementação deste projeto, mais famílias possam se sentir seguras em suas casas, reduzindo a incidência de invasões domiciliares e melhorando a qualidade de vida dos moradores. O desenvolvimento de tecnologias de segurança acessíveis é um passo crucial para enfrentar os desafios impostos pela crescente criminalidade e para promover um ambiente mais seguro para todos.

## Trabalhos futuros

Como possibilidade de trabalho futuro para este estudo, está a adoção de outras bibliotecas de captura e análise de imagens com mais funcionalidades e recursos que a usada no projeto atualmente, como a biblioteca YOLOv8, que suporta a detecção, segmentação, classificação e rastreamento de imagens, usando técnicas com aprendizado profundo e redes neurais artificiais.

# REFERÊNCIAS

# Bibliografia

Alves, G. (13 de Outubro de 2020). *Iaexpert Academy*. Acesso em 20 de Maio de 2024, disponível em IA Expert Academy: https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/?doing\_wp\_cron=1717502734.4242470264434814453125

Antonello, R. (2017). *Introdução a Visão Computacional com Python e OpenCV.*

(2023). *Anuário Brasileiro de Segurança Pública .* Fonte: FBSP: https://publicacoes.forumseguranca.org.br/items/6b3e3a1b-3bd2-40f7-b280-7419c8eb3b39

Carolina Monard, M., & Augusto Baranauskas, J. (2003). *Conceitos sobre Aprendizado de Maquina.*

*Docs opencv org*. (2024). Acesso em 20 de Fevereiro de 2024, disponível em OpenCV - Open Computer Vision Library: https://docs.opencv.org/4.x/d6/d00/tutorial\_py\_root.html

*Docs python org*. (2024). Acesso em 5 de Maio de 2024, disponível em Python Docs: https://docs.python.org/3/library/tkinter.messagebox.html

*Docs python org*. (2024). Acesso em 27 de Fevereiro de 2024, disponível em Python Docs: https://docs.python.org/3/library/webbrowser.html

*Docs python org*. (2024). Acesso em 20 de Maio de 2024, disponível em Python Docs: https://docs.python.org/3/library/sys.html

*Docs python org*. (2024). Acesso em 20 de Maio de 2024, disponível em Python Docs: https://docs.python.org/3/library/subprocess.html

*Docs python org*. (2024). Acesso em 22 de Fevereiro de 2024, disponível em Python Docs: https://docs.python.org/3/library/os.html

*Docs python org*. (2024). Acesso em 22 de Fevereiro de 2024, disponível em Python Docs: https://docs.python.org/3/library/time.html

*Espressif*. (2024). Acesso em 21 de Maio de 2024, disponível em Espressif Systems: https://www.espressif.com/en/products/socs/esp32

Kazuhiro Okabe, R., & AntonioCarro, S. (2014). *RECONHECIMENTO FACIAL EM IMAGENS CAPTURADAS POR CÂMERAS DIGITAIS DE REDE.*

Marim, Y. V. (2019). *DETECÇÃO DE OBJETOS: ESTUDO E APLICAÇÃO DA ARQUITETURA R-CNN.*

*opencv*. (2024). Acesso em 18 de Fevereiro de 2024, disponível em OpenCV: https://opencv.org/

*Pillow readthedocs io*. (2024). Acesso em 17 de Maio de 2024, disponível em Pillow: https://pillow.readthedocs.io/en/stable/

*pypi*. (2024). Acesso em 2 de Maio de 2024, disponível em PyPI: https://pypi.org/project/customtkinter/0.3/

*Pypi org*. (22 de Fevereiro de 2024). Fonte: PyPI: https://pypi.org/project/pywhatkit/

*Python*. (2024). Acesso em 5 de Fevereiro de 2024, disponível em Python: https://www.python.org/

Stringhini, D., & Marengoni, M. (2009). *Tutorial: Introdução à Visão Computacional usando OpenCV.*

*Timgolden me uk*. (2020). Acesso em 22 de Fevereiro de 2024, disponível em Tim Golden's Stuff: https://timgolden.me.uk/pywin32-docs/

Wangenheim, A. v. (2018). *Lapix Ufsc*. Acesso em 24 de Maio de 2024, disponível em LAPIX - Image Processing and Computer Graphics Lab - UFSC: https://lapix.ufsc.br/ensino/visao/visao-computacionaldeep-learning/deteccao-de-objetos-em-imagens/

*Whatsapp*. (2024). Acesso em 20 de Maio de 2024, disponível em WhatsApp: https://www.whatsapp.com/?lang=pt\_BR

Zuba, F., & Paula Proença, A. (2023 de Dezembro de 2023). *G1 Globo*. Acesso em 23 de Abril de 2024, disponível em G1 - O portal de notícias da Globo: https://g1.globo.com/mg/minas-gerais/noticia/2023/12/28/numero-de-invasoes-de-residencias-cresce-em-minas-gerais-e-belo-horizonte-em-2023.ghtml

# APÊNDICE