

JOÃO PEDRO TEIXEIRA DE OLIVEIRA

**DOMOTICA COMO AUXÍLIO E  
SEGURANÇA PARA  
PESSOAS COM SURDEZ**

Projeto de Trabalho de Conclusão de  
Curso da Engenharia de Controle e  
Automação da Universidade Federal de  
Uberlândia - UFU - Câmpus Santa  
Mônica, como requisito para a obtenção  
do título de Graduação em Engenharia  
de Controle e Automação

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Elétrica

Orientador: Prof. Dr. Fábio Vincenzi

**UBERLÂNDIA**

**2023**

## **AGRADECIMENTOS**

Primeiro agradeço a Deus pela vida e todo cuidado em todos os momentos da minha vida.

Aos meus pais que sempre me apoiaram, incentivaram e sempre me ajudaram a superar os desafios da vida. Ao meu irmão pela companhia e todos os momentos alegres.

Ao meu orientador, pela paciência, auxílio e suporte durante a realização desse projeto.

## Resumo

A automação residencial, juntamente com a Internet das Coisas (IoT), procura melhorar e controlar ambientes, fornecendo assim uma melhor qualidade de vida para as pessoas. A união da automação com tecnologias de acessibilidade permite melhorias significativas para a vida das pessoas com deficiência, podendo transformar a vida do indivíduo, suas interações e percepções do mundo ao seu redor, proporcionando uma maior autonomia e qualidade de vida. Nesse contexto, foi desenvolvido nesse trabalho um dispositivo que buscou contribuir para a automação de acessibilidade.

O dispositivo proposto foi um interfone adaptado, o qual realizará a integração de um botão, uma câmera IP e o celular do usuário. O sistema buscou o seguinte resultado, sempre que o botão é acionado, o sistema enviará um comando para ativar a vibração do celular do usuário. Isso permite que o usuário identifique quem está acionando o botão e tome a decisão de permitir ou não a liberação da trava eletrônica através do aplicativo em seu celular.

O trabalho foi dividido em três grandes partes, um referencial teórico, onde são fornecidas informações e conceitos necessários para a compreensão do projeto, metodologia, onde são apresentados panoramas gerais do projeto e como foram construídas e idealizadas partes fundamentais do dispositivo, e por fim os resultados que possuem uma amostra do dispositivo feito.

**Palavras-chave:** Automação Residencial, Software, Acessibilidade.

## Abstract

Home automation, together with the Internet of Things (IoT), seeks to improve and control environments, thus providing a better quality of life for people. The union of automation with accessibility technologies allows significant improvements to the lives of people with disabilities, being able to transform the individual's life, their interactions and perceptions of the world around them, providing greater autonomy and quality of life. In this context, a device was developed in this work that sought to contribute to accessibility automation.

The proposed device was an adapted intercom, which will integrate a button, an IP camera and the user's cell phone. The system sought the following result, whenever the button is pressed, the system will send a command to activate the vibration of the user's cell phone. This allows the user to identify who is pressing the button and make the decision to allow or not to release the electronic work through the application on their cell phone.

The work was divided into three large parts, a theoretical framework, where information and concepts necessary for understanding the project are provided, methodology, where general overviews of the project are presented and how fundamental parts of the device were built and idealized, and finally the results that have a sample of the device made.

Keywords: Home Automation, Software, Aecessibility

## Lista de ilustrações

Figura 1 – Exemplo de um Microcontrolador ESP32.....	12
Figura 2- Flow NodeRed 14.....	14
Figura 3- Infraestrutura da comunicação MQTT.....	16
Figura 4- Resumo das QoS MQTT.....	18
Figura 5- Interface Android Studio.....	20
Figura 6- Arquitetura do projeto.....	22
Figura 7- Criação de um novo usuário.....	23
Figura 8- Código Notificação e Vibração.....	24
Figura 9- Código para publicação no tópico.....	26
Figura 10- Código construção reprodutor de vídeo VLC.....	27
Figura 11- Circuito de registro e acionamento.....	28
Figura 12- Flow para tratamento de comandos.....	29
Figura 13- Telas de Login e Cadastro do Aplicativo.....	31
Figura 14- Tela Principal do Aplicativo.....	32
Figura 15- Notificação de novo visitante.....	33
Figura 16- Tela de Monitoramento.....	34
Figura 17- Circuito real com Esp32 e Relé.....	35

Figura 18- Circuito desligado e acionado.....36

## Lista de abreviaturas e siglas

bps	Bits por segundo
<i>Clock</i>	Frequência de um processador
GHz	GigaHertz - Unidade de medida de frequência
Hz	Hertz - Unidade de medida de frequência
<i>IDE</i>	Ambiente de desenvolvimento Integrado
IHM	Interface Homem Máquina
IoT	<i>Internet of Things</i>
JAVA	Linguagem de Programação
Mbps	Megabit por segundo
MQTT	<i>Message Queueing Telemetry Transfer</i>
QoS	<i>Quality of Service</i>
Rx	<i>Final de recebimento</i>
TCP	<i>Transmission Control Protocol</i>
Tx	<i>Final de envio</i>
UDP	<i>User Data Protocol</i>
XML	<i>Extensible Markup Language</i>

## Sumário

<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>10</b>
	<b>1.1 Justificativa.....</b>	<b>10</b>
	<b>1.2 Objetivo .....</b>	<b>12</b>
<b>2.</b>	<b>REFERENCIAL TEÓRICO.....</b>	<b>13</b>
	<b>2.1 Estado da Arte .....</b>	<b>13</b>
	<b>2.2 Microcontrolador .....</b>	<b>14</b>
	<b>2.3 Node-RED .....</b>	<b>16</b>
	<b>2.4 Protocolo MQTT .....</b>	<b>18</b>
	<b>2.5 ANDROID STUDIO.....</b>	<b>21</b>
	<b>2.6 Arduino IDE .....</b>	<b>23</b>
<b>3</b>	<b>METODOLOGIA .....</b>	<b>24</b>
	<b>3.1 Etapa 1 : Codificação do Aplicativo Android.....</b>	<b>25</b>
	<b>3.2 Etapa 2 : Codificação da IDE Arduino e montagem do circuito</b>	

	<b>3.3 Etapa 3 : Criação do Flow no Node-Red .....</b>	<b>31</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES .....</b>	<b>33</b>
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>39</b>

## **1. INTRODUÇÃO**

Existem diversos níveis de surdez, sendo elas, leve, moderada, profunda e total, e em todos os níveis existem impactos na vida do portador, da sua família e em toda a sociedade (MOREIRA, 2023) . Segundo o IBGE em 2022 existiam 2,3 milhões de pessoas com algum grau de surdez no Brasil, e de acordo com World Health Organization (2021) 1,5 bilhões de pessoas com alguma deficiência auditiva no mundo. A surdez, independentemente do seu grau muda a forma com que o indivíduo percebe o mundo, interage com seu meio, com seus familiares e com as esferas sociais no qual está inserido.

Diversos estudos são realizados anualmente buscando formas de inclusão e combate a surdez, uma das formas encontradas para contrapor esse problema é a automação residencial.

Domótica é o termo utilizado quando está sendo tratado assuntos sobre automação residencial. Com a utilização de recursos tecnológicos, como o crescente uso de dispositivos IoTs, aplicados nas residências é possível transformar o ambiente e o cotidiano de forma mais agradável e inclusiva, em 2023 o uso de dispositivos de IoT deve crescer em até 20% (ZANATTA, 2021). A integração de novas tecnologias proporciona que alguns processos diários sejam automatizados, bem como facilitados, permitindo assim que algumas tarefas sejam realizadas com mais autonomia e inclusão.

Proporcionando uma maior qualidade de vida e autonomia por meio da domótica, pessoas com diferentes níveis de surdez conseguem realizar tarefas diárias sem serem impactadas pela limitação auditiva. Nesse contexto o presente trabalho propõe o desenvolvimento aparelho de controle de fluxo para auxílio e incremento da segurança para pessoas surdas.

### **1.1 Justificativa**

Este projeto se baseia na motivação de se aprimorar e buscar maiores conhecimentos das tecnologias voltadas à automação residencial. Isso decorre do reconhecimento de como essas tecnologias têm o potencial de aprimorar

significativamente a qualidade de vida da população. A justificativa central para esta iniciativa reside na compreensão de que a automação residencial não apenas simplifica tarefas diárias, mas também proporciona uma maior inclusão social.

Em meio a um mercado que continua sua expansão constante, inúmeros projetos estão sendo conduzidos com o objetivo de oferecer soluções cada vez mais eficazes aos clientes, visando não apenas a redução de custos, mas também a melhoria significativa na qualidade de vida. Conforme indicado por New Voice (2022), "O mercado de automação residencial está projetado para experimentar um crescimento notável, passando dos estimados US\$ 72,30 bilhões em 2021 para impressionantes US\$ 163,24 bilhões em 2028."

Dentro deste contexto de mercado em ascensão, as tecnologias assistivas estão em contínuo desenvolvimento, e este trabalho se motiva na realização da solução de um problema de acessibilidade, visando também uma redução nos custos do produto. De acordo com Radabaugh (2023), "Para a maioria das pessoas, a tecnologia simplifica as tarefas. Mas, para pessoas com deficiência, a tecnologia torna coisas que eram antes inalcançáveis em possibilidades reais." Portanto, torna-se imperativo que esses dispositivos sejam acessíveis a indivíduos com qualquer grau de deficiência, assegurando que a inovação tecnológica beneficie a todos.

## 1.2 Objetivo

O objetivo central deste projeto abrange a criação de um protótipo acessível e de operação descomplicada, com o intuito de oferecer suporte a indivíduos com deficiência auditiva. O protótipo proporciona que o indivíduo possa visualizar e controlar o acesso a sua residência de forma eficiente, por meio da integração de uma campanha com seu aparelho Android, seja ele um celular ou um Tablet. O dispositivo realiza a notificação de novos visitantes por meio de notificações sensoriais visuais e sensíveis, através de notificações e vibração do aparelho celular do usuário.

Essa integração proporciona um aumento da independência do deficiente auditivo, também promovendo uma maior segurança e tranquilidade em relação a sua própria residência.

## 2. REFERENCIAL TEÓRICO

Nessa seção foram colocadas referências teóricas para realização e compreensão desse trabalho, abordando explicações sobre o funcionamento das ferramentas e protocolos que foram utilizados no trabalho.

### 2.1 Estado da Arte

Freitas e Silva (2016, p. 1) observaram em seus estudos, “É a partir da tecnologia desenvolvida ao longo do tempo que pode ser vislumbrado um avanço em uma das características mais humanas: a solidariedade”.

Os avanços tecnológicos, ao longo dos anos, têm se traduzido em soluções que visam melhorar a qualidade de vida e a inclusão de indivíduos com deficiências. Essas inovações abrem novos horizontes e oportunidades para a acessibilidade e a autonomia de pessoas com deficiência, abrangendo uma variedade de domínios, como mobilidade, comunicação, educação entre outros.

Em 2016 foi proposto por Freitas e Silva, um projeto de campanha para deficientes auditivos, utilizando um Arduino, um Buzzer, um smartphone. Porém, diferente da proposta deste trabalho, a comunicação utilizada foi o protocolo de transmissão de dados Bluetooth e comunicação serial, também não houve a presença de dispositivos para monitoramento e acionamento de qualquer dispositivo de fechadura.

Almeida Júnior (2016) propôs a criação de um dispositivo que permite realizar o reconhecimento de eventos sonoros no ambiente, e em um display mostrar o que está acontecendo. O objetivo do projeto foi auxiliar de forma visual o deficiente auditivo a obter maior conforto no ambiente ao seu redor.

Um trabalho foi idealizado por Silva *et al.* (2022), de uma campanha adaptada para pessoas com surdez, onde uma pulseira vibra através de um vibracall ao se acionar o botão do interfone. É importante ressaltar que mesmo os projetos apresentando semelhanças, na proposta atual foi utilizado um smartphone como dispositivo de notificação, além disso, um aplicativo foi criado para integrar diversas funcionalidades, incluindo a capacidade de monitorar a residência por meio de uma

câmera de segurança, o sistema de notificação de visitantes e a capacidade de controlar o acesso às fechaduras. Essas adições significativas ampliam consideravelmente a utilidade e a eficiência do sistema, proporcionando não apenas uma solução de campanha acessível, mas também um sistema de segurança residencial mais completo

Ainda no âmbito de desenvolvimento de tecnologias assistivas, por Pureza (2017), foi desenvolvido um dispositivo de alerta para crianças surdas, buscando proporcionar uma maior autonomia para as crianças durante suas atividades em casa, diminuindo assim sua dependência.

## **2.2 Microcontrolador**

Os microcontroladores são circuitos integrados que incorporam diversas características encontradas em computadores IBM PC, porém, notavelmente, são caracterizados por seu tamanho compacto e uma arquitetura baseada na tecnologia ARM (MELLO, 2022).

Esses dispositivos englobam componentes fundamentais, tais como núcleo de processamento, memórias RAM/ROM/FLASH, geradores de clock, conversores e interfaces de entrada e saída.

Complementando sua versatilidade, os microcontroladores podem ser programados para desempenhar tarefas específicas (CUNHA, 2007). Apresentam dimensões limitadas e uma significativa redução no custo em comparação com os tradicionais PCs IBM. Além disso, oferecem uma eficiência energética superior e despesas de fabricação mais baixas. Essas características tornam os microcontroladores populares em uma ampla gama de sistemas embarcados. Desta forma, se torna possível aplicar o microcontrolador em controles remotos, eletrodomésticos, controles de carros, brinquedos e outros tantos dispositivos automatizados. (MELLO, 2022).

À medida que o conceito de Internet das Coisas (IoT) ganha destaque, a utilização de microcontroladores com conectividade sem fio vem se tornando cada vez mais comum. Isso se deve à facilidade de monitoramento, à capacidade de conexão à rede e ao aprimoramento da segurança.

Um exemplo notável de controlador com conectividade sem fio que se

destaca no mercado é o ESP8266. Este dispositivo é amplamente utilizado devido às suas características, que incluem conectividade wireless (802.11 b/g/n), faixa de frequência de 2,4 GHz a 2,5 GHz, antena embutida, interface Serial UART, SPI Flash de 8Mbits, 50KB de RAM, suporte para comunicação TCP e UDP e um alcance de aproximadamente 90 metros (CURVELLO, 2015).

O ESP8266-07, em particular, é notável pelo seu tamanho reduzido, o que permite maior flexibilidade em montagens. Além disso, ele já possui funcionalidade de conexão à rede Wi-Fi. Verificando seu baixo valor de mercado, quando comparado com controladores mais populares, como Arduino ou Raspberry Pi, o ESP8266 é uma escolha viável para projetos de baixo custo e projetos de implementação em dispositivos inteligentes. Para aplicações mais robustas, o ESP32 surge como uma alternativa.

O ESP32, desenvolvido pela Espressif, é adequado para aplicações mais exigentes e apresenta características como suporte para temperaturas extremas de -40°C a 125°C, baixo consumo de energia, conectividade wireless (802.11 b/g/n), antena embutida, funcionalidade Bluetooth e interfaces UART, SPI, SDIO e I2C (ESPRESSIF, 2023).

No contexto deste trabalho, a escolha recaiu sobre o ESP32 devido à sua robustez e eficiência energética, tornando-o uma opção ideal para a aplicação em desenvolvimento. Seu preço de mercado semelhante ao do ESP8266-07 reforça sua viabilidade como uma solução eficaz. Na figura 1 está presente o exemplo de uma placa de desenvolvimento ESP32, que realiza a integração do microcontrolador com outros componentes, a qual busca facilitar o desenvolvimento de protótipos utilizando microcontroladores.

Figura 1 Exemplo de uma placa de desenvolvimento ESP32



Fonte: <https://www.makehero.com/produto/modulo-wifi-esp32-bluetooth/> Acesso em:

10/10/2023

## 2.3 Node-RED

O Node-Red se apresenta como uma valiosa ferramenta de código aberto, proporcionando um ambiente propício para a conexão de dispositivos pertencentes à Internet das Coisas (IoT). Sua funcionalidade é efetivada por meio de módulos previamente disponibilizados na aplicação, o que se traduz em uma abordagem ágil na criação de canais de comunicação entre uma diversidade de dispositivos.

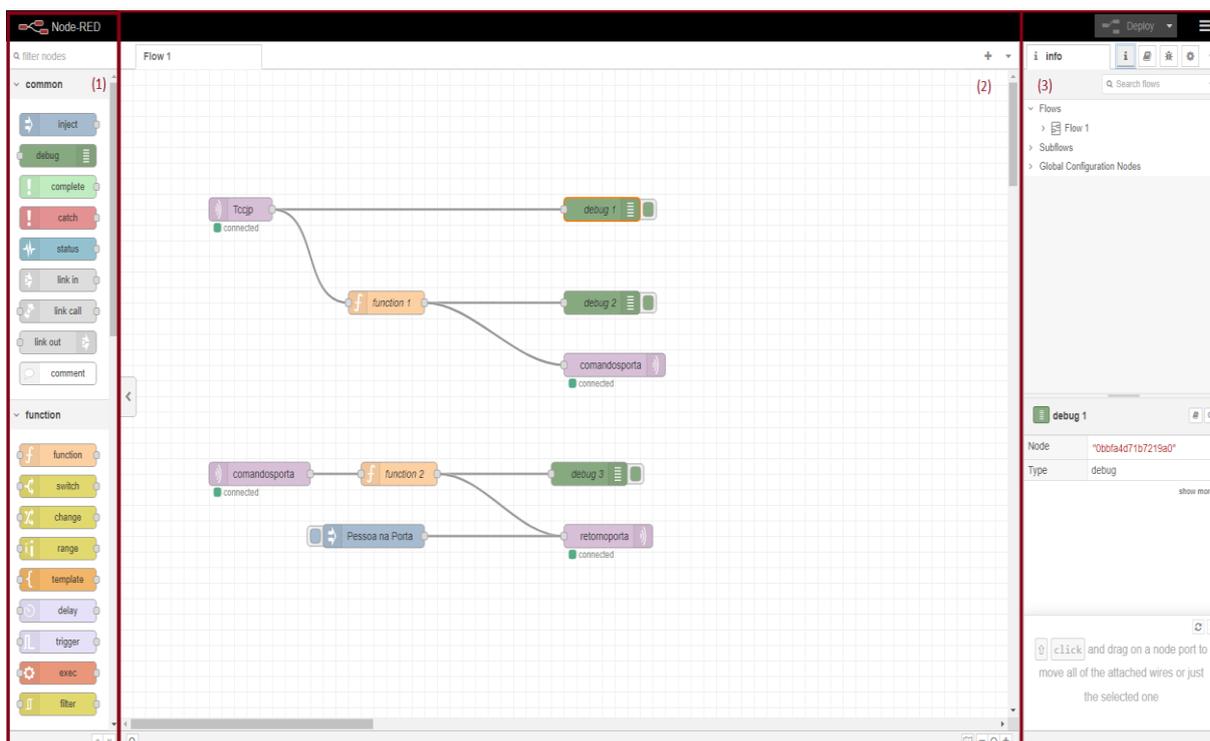
A escrita de scripts para o Node-Red se baseia na linguagem Node.js e a interação com a plataforma é realizada de modo visual e intuitivo. É necessário que o usuário arraste os blocos, ou nós, para a tela e os conecte (SILVA JÚNIOR, 2020).

Ademais, além da aplicação na nuvem, o Node-Red oferece o Node-Red Arduino, um componente que se destina a estabelecer comunicações com placas conectadas via conexão serial, permitindo um diálogo full duplex.

É relevante destacar que o Node-Red admite a adição de diversos "nós," ou blocos de funcionalidades, ampliando, assim, sua versatilidade e adaptabilidade para distintas aplicações. No entanto, no escopo deste projeto em particular, optou-se por utilizar os nós padrões já fornecidos pela plataforma, atendendo às necessidades específicas da situação sem a necessidade de incorporar novos elementos.

O Node-Red, com suas características de código aberto e sua capacidade de conexão com protocolos de comunicação de dispositivos IoT, tornam a plataforma um meio eficiente nas soluções de automação residencial, tornando-as acessíveis e customizáveis para diversos cenários e necessidades.

Figura 2 Flow NodeRed



Fonte: Autoria própria

Ao examinar a Figura 2, observa-se como é a interface do NodeRed, em (1) estão localizados os blocos com diferentes funções, como o bloco de conexão MQTT, o bloco de função, onde são realizadas programações em JavaScript, e o bloco de debug, todos presentes na imagem, em (2) é realizada a montagem de um fluxo de trabalho, observa-se que tarefas complexas são realizadas de forma rápida através da ligação de blocos e em (3) podem ser visualizadas informações pertinentes de cada bloco. Através dessa ilustração, é possível compreender o funcionamento do Node-Red e como essa plataforma simplifica a comunicação entre diversos dispositivos.

A análise do fluxo revela que o Node-Red atua como uma espécie de ponte facilitadora, tornando mais eficiente o processamento da comunicação MQTT. Essa plataforma desempenha um papel fundamental na filtragem de mensagens e no encaminhamento das mesmas para os dispositivos apropriados de forma direta e descomplicada. Isso demonstra o valor do Node-Red como uma ferramenta que aprimora a gestão e a distribuição de mensagens em uma rede de dispositivos, contribuindo para uma comunicação eficaz e confiável em ambientes de automação

residencial e IoT.

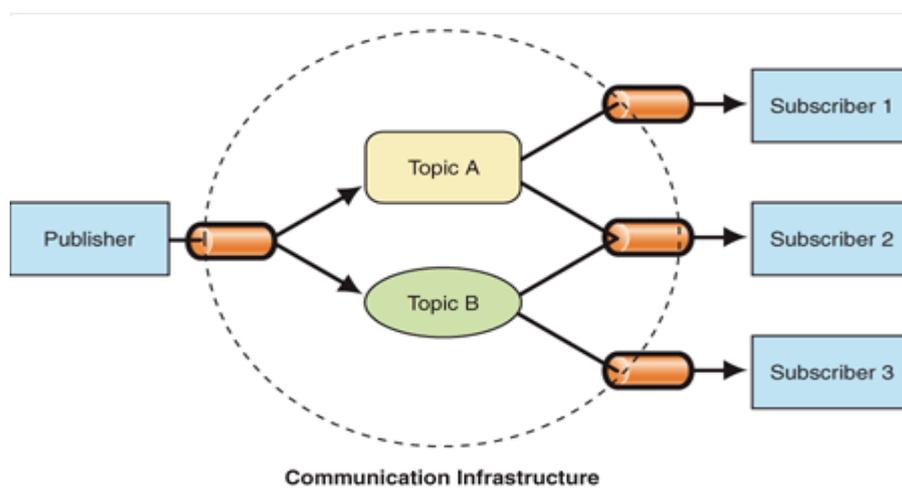
## 2.4 Protocolo MQTT

O Protocolo MQTT, também conhecido como Message Queuing Telemetry Transport, representa uma solução de comunicação de código aberto que se destaca por sua leveza e natureza assíncrona. Ele foi concebido com a finalidade de simplificar a eficiente troca de mensagens entre dispositivos. A origem desse protocolo remonta ao seu uso inicial na transmissão de dados de telemetria de oleodutos via satélite. Hoje, o protocolo é amplamente adotado no contexto da Internet das Coisas, por possuir um baixo consumo de banda e energia.

O funcionamento do Protocolo de comunicação MQTT é baseado no modelo publisher-subscriber, nesse modelo de comunicação, um publisher transmite uma mensagem para um tópico específico, então o subscriber se inscreve neste mesmo tópico e, como resultado, passa a receber as mensagens enviadas pelo publicador. Esse modelo de operação busca estabelecer uma comunicação eficaz e flexível entre dispositivos em uma rede, uma vez que o mesmo dispositivo pode ser Publisher e Subscriber (SOUZA, 2019).

Portanto, o Protocolo de comunicação MQTT se destaca como uma solução versátil que desempenha um papel significativo na capacitação da comunicação em ambientes de IoT, tornando a troca de dados mais eficiente e com baixo consumo de rede. Os clientes MQTT podem ser executados em vários dispositivos, de sensores e atuadores a dispositivos hand held e sistemas de veículos (IBM,2021). Na figura 3 é verificada uma representação visual da infraestrutura da comunicação através do protocolo MQTT.

Figura 3: Infraestrutura da comunicação MQTT



Fonte: <https://hackernoon.com/observer-vs-pub-sub-pattern-50d3b27f838c> Acesso em: 13/09/2023

É importante destacar que a interação entre o publicador e o assinante no protocolo, não requer que ambos estejam na mesma rede ou conectados diretamente via serial. Em vez disso, a conexão e o roteamento das mensagens são intermediados por um componente central, denominado Broker.

O Broker MQTT, também conhecido como servidor MQTT, desempenha um papel crucial em um sistema de comunicação MQTT. Sua função principal é servir como um intermediário entre o publicador e o assinante (Conceição, Costa, 2019). No Broker estão presentes os tópicos MQTT, ao receber as mensagens de um cliente, ele se encarrega de direcioná-las para os destinatários apropriados. Além disso, o Broker assume a responsabilidade de garantir que as mensagens sejam entregues com êxito aos locais designados, independentemente do número de conexões simultâneas ou do estado de disponibilidade do destinatário no momento da entrega.

Para garantir a entrega eficaz das mensagens a todos os destinatários, independentemente do número de conexões envolvidas, o Broker oferece suporte à criação de múltiplos tópicos que podem funcionar simultaneamente (Conceição, Costa, 2019). Além disso, ele possibilita que vários clientes assinem o mesmo tópico, o que amplia a flexibilidade do sistema. É importante mencionar que existe uma ampla variedade de Brokers MQTT disponíveis para utilização, sendo que alguns dos mais comuns incluem o Eclipse Mosquitto, que é uma opção de código aberto, o HiveMQ, geralmente empregado em soluções empresariais, e o Paho MQTT, que é frequentemente utilizado em aplicações móveis.

Dentro do contexto da comunicação MQTT, além do papel crucial desempenhado pelo broker, há um elemento adicional de extrema importância denominado Qualidade de Serviço (QoS). A Qualidade de Serviço refere-se a um conjunto de níveis de garantia que controlam a entrega de mensagens entre os clientes MQTT. Esses níveis de QoS, três no total, desempenham um papel fundamental na determinação do grau de confiabilidade e garantia de entrega das mensagens.

O primeiro nível de QoS é conhecido como QoS 0, ou nível de entrega mínima. Nesse nível, a mensagem é enviada apenas uma vez para o receptor, e o

Broker MQTT encaminha imediatamente qualquer nova mensagem aos clientes assinantes do tópico, porém, nesse nível, não existe qualquer garantia ou confirmação de que a mensagem tenha sido efetivamente recebida pelo cliente. Este é o nível mais ágil e leve dos três, uma vez que não envolve recursos de confirmação.

O segundo nível é o QoS 1, ou nível de entrega pelo menos uma vez. Nesse cenário, a mensagem do publicador é enviada ao Broker MQTT, que a encaminha para os assinantes do tópico. Ao receber a mensagem, o assinante envia uma confirmação de recebimento, conhecida como puback, de volta ao publicador. Caso não haja confirmação de recebimento, o broker realiza uma requisição para o Publisher, enviando novamente a mensagem, esse processo se repete até que seja obtida a confirmação puback, esse tipo de tratamento pode gerar múltiplas requisições e múltiplos pubacks, e caso a aplicação não esteja preparada para lidar com essas mensagens pode gerar problemas no desempenho da aplicação.

O terceiro nível é o QoS 2, ou entrega exatamente uma vez. Nesse cenário, o publicador envia uma mensagem para o Broker MQTT, que a encaminha para os assinantes do tópico. Ao receber a mensagem de confirmação de recebimento, chamada de pubrec, o publicador envia uma mensagem de conclusão da comunicação, conhecida como pubcomp. Esse processo garante que a mensagem seja entregue exatamente uma única vez e que não se perca durante o processo. Embora o QoS 2 seja o mais confiável dos três níveis, ele também é o mais pesado em termos de tráfego de rede, devido à necessidade de confirmações adicionais.

O protocolo MQTT, apresenta uma série de vantagens notáveis em comparação com outros protocolos de comunicação, especialmente quando aplicado ao contexto da Internet das Coisas (IoT). Entre essas vantagens, destacam-se características que o tornam altamente adequado para dispositivos IoT.

Uma das vantagens mais proeminentes é a eficiência de recursos oferecida pelo MQTT. Ele se destaca por exigir um baixo consumo de largura de banda e de capacidade de processamento, tornando-o ideal para dispositivos IoT, que frequentemente operam em ambientes com recursos limitados. Essa eficiência assegura que os dispositivos IoT possam se comunicar de maneira eficaz, sem sobrecarregar a rede ou esgotar os recursos de processamento, o que é fundamental para a operação contínua e estável desses dispositivos.

A estrutura Publicador - Assinante não apenas fornece uma organização

eficaz das mensagens, mas também torna o processo de expansão das redes IoT mais simples e sem complicações significativas. À medida que novos dispositivos são adicionados à rede, basta que eles se inscrevam nos tópicos relevantes, sem a necessidade de redesenhar toda a arquitetura da rede. Isso permite que os projetos IoT cresçam rapidamente e se adaptem às mudanças nas necessidades de comunicação, promovendo a expansão ágil e eficiente das redes IoT.

## **2.5 ANDROID STUDIO**

O Android Studio, um ambiente de Desenvolvimento Integrado (IDE), que possibilita uma maior velocidade na criação de aplicativos móveis para Android. Este software é oferecido gratuitamente pela Google. Uma das principais características do Android Studio é seu forte suporte à integração com os diversos serviços também oferecidos pela Google, o que simplifica a incorporação de diversas funcionalidades nos aplicativos (HELLMANN, 2016).

No que diz respeito às linguagens de programação, a IDE suporta tanto Kotlin quanto Java como opções nativas. A escolha entre as duas depende da preferência do desenvolvedor e dos requisitos do projeto. Neste trabalho específico, a decisão foi adotar a linguagem Java para a codificação do aplicativo.

Um dos recursos notáveis do Android Studio é a capacidade de programar a interface visual do aplicativo de maneira intuitiva. Os desenvolvedores podem utilizar blocos de funções prontas para agilizar o desenvolvimento ou optar por codificar manualmente em XML, proporcionando um alto grau de flexibilidade na criação da interface do usuário.

Além disso, o Android Studio suporta uma integração nativa com os serviços Google, como o Google Firebase e o Google Cloud, que oferecem recursos robustos para gerenciamento de dados, controle de tráfego e segurança dos dados dos usuários. Isso é fundamental para garantir a confiabilidade e a segurança dos aplicativos desenvolvidos.

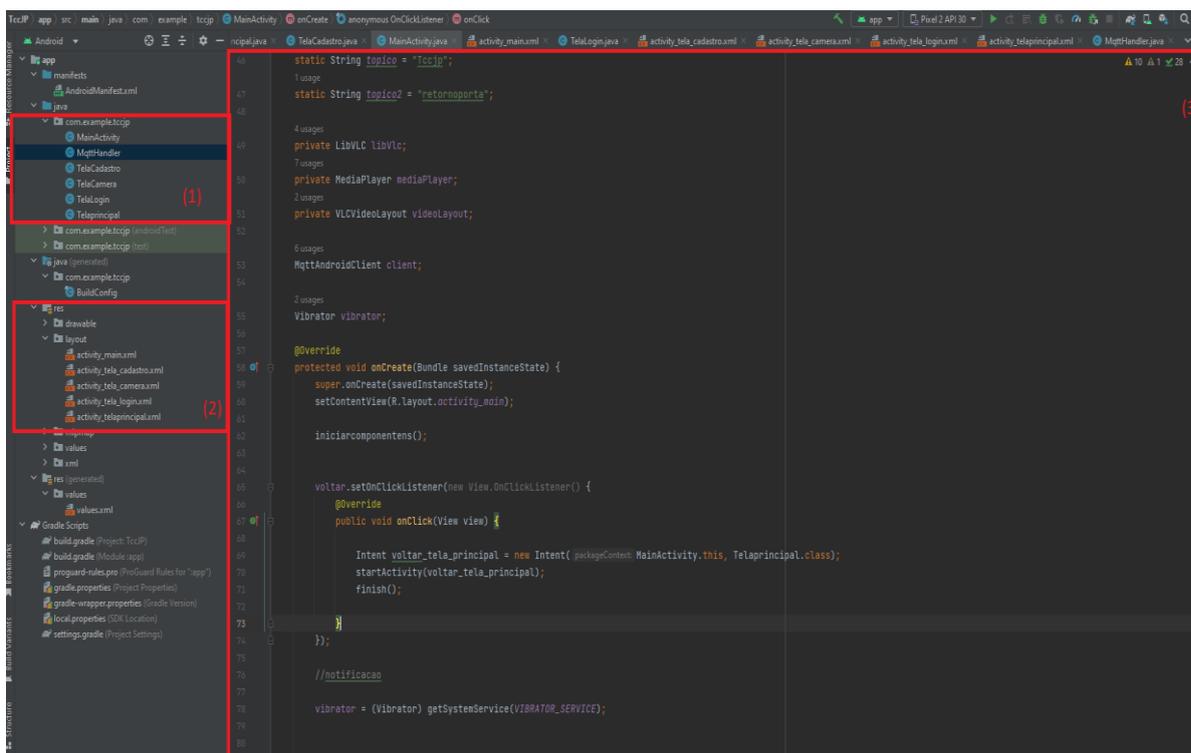
Na figura 4, pode-se observar exemplo do ambiente de desenvolvimento do aplicativo. A primeira seção, identificada como (1), é reservada para a criação de cada tela do aplicativo. Dentro desta seção, são gerados arquivos específicos para cada tela, contribuindo para uma organização mais eficaz do projeto. Essa abordagem ajuda a manter a estrutura do aplicativo bem definida e simplifica a

manutenção.

A segunda seção, marcada como (2), é destinada à criação do layout das telas do aplicativo. Assim como na seção anterior, são criados arquivos individuais para cada tela. Além disso, nesta etapa, ocorre a integração entre os elementos visuais e os arquivos de codificação criados na seção (1). Isso significa que as interfaces visuais podem ser projetadas e modificadas sem afetar os arquivos de codificação, garantindo uma maior flexibilidade no desenvolvimento.

Por último, na terceira seção, identificada como (3), ocorre a codificação real do aplicativo. Neste espaço, os desenvolvedores podem escrever o código-fonte necessário para implementar a lógica e as funcionalidades do aplicativo. Além disso, essa seção oferece a capacidade de realizar depuração (debug) do código, o que facilita a identificação e resolução de problemas e erros no código-fonte

Figura 4 : Interface Android Studio



Fonte: Autoria própria

## 2.6 Arduino IDE

O ambiente de desenvolvimento integrado Arduino, frequentemente chamado de Arduino IDE, é uma plataforma composta por um editor de texto que estabelece uma interface entre o computador do usuário e o hardware dos microcontroladores Arduino e outras placas similares, aproveitando-se de bibliotecas adicionais que aprimoram sua funcionalidade (Oliveira, 2017).

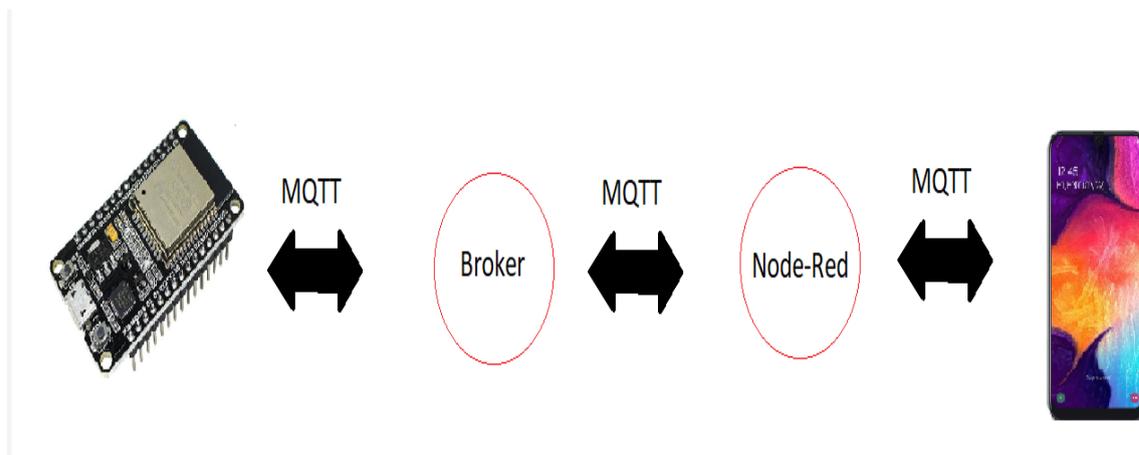
O editor de texto oferece diversas ferramentas projetadas para facilitar a criação de protótipos. Isso inclui uma verificação do código gerado pelo usuário e um console que detalha possíveis erros no código, simplificando, assim, a correção das instruções. Além disso, a IDE proporciona uma função que permite a monitoração de informações por meio da conexão serial, tornando possível a criação de variáveis para a depuração do código, as quais podem ser visualizadas no monitor serial do software.

A utilização desse ambiente é relativamente simples. O código que será compilado deve ser inserido dentro de duas funções predefinidas pela Arduino IDE: "setup" e "loop." A função "setup" é executada apenas uma vez e tem a finalidade de realizar configurações prévias essenciais ao código. Já a função "loop" é executada repetidamente enquanto o microcontrolador estiver alimentado, possibilitando a execução contínua das ações programadas. Esse ambiente fornece uma base sólida para o desenvolvimento de projetos, permitindo que os usuários criem e testem protótipos de forma eficaz e simplificada.

### 3 METODOLOGIA

Foi realizado um projeto de automação residencial, utilizando ferramentas Android Studio, Mosquitto Broker e Node-Red, bem como protocolo de comunicação MQTT e um microcontrolador ESP32. A arquitetura do projeto encontra-se na figura 5.

Figura 5 : Arquitetura do projeto



Fonte: Autoria própria

A automação proposta, consiste em realizar a abertura de uma trava eletrônica por meio de um aplicativo de celular, o qual realiza a integração do atuador da trava, e de uma câmera de segurança, possibilitando assim ao usuário verificar quem está na porta antes de abri-la, ou caso seja necessário pedir ajuda para situações de risco.

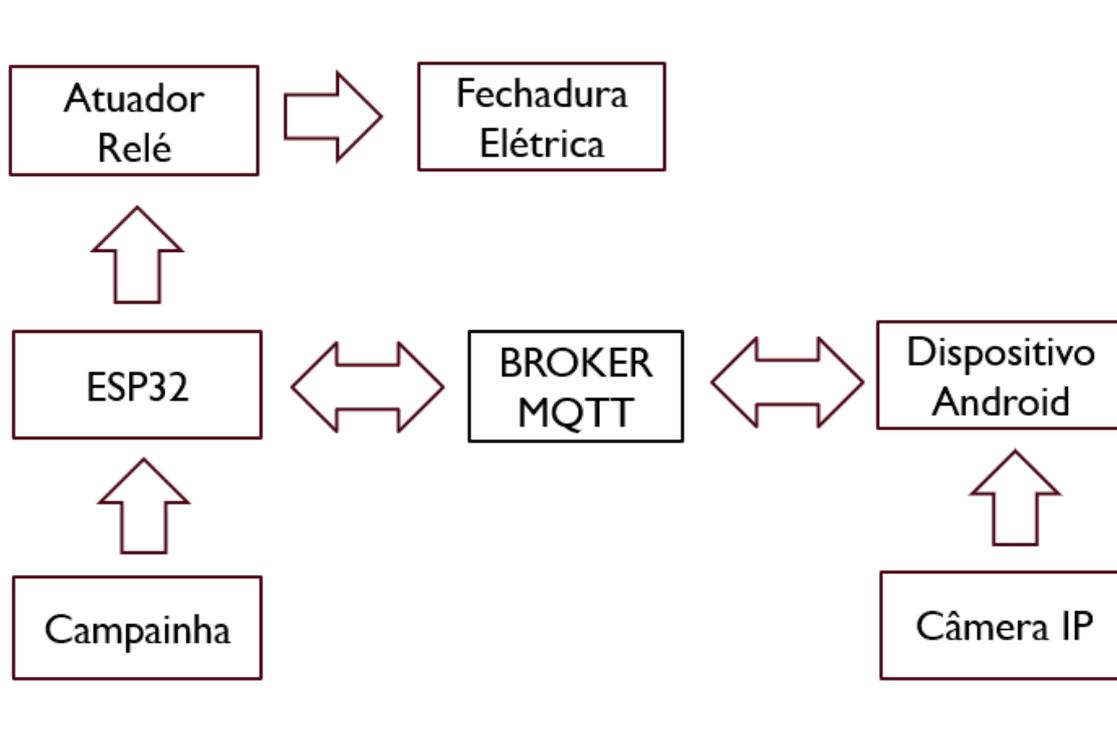
A comunicação entre o atuador, e o aplicativo é realizada através do protocolo de comunicação MQTT. O controle de chegada de novas pessoas, e abertura da trava é realizado pelo microcontrolador ESP32, o qual solicita autorização para o dispositivo Android do usuário antes de realizar as ações.

Por se tratar de um sistema de segurança, e que está presente em uma rede de automação residencial, foi escolhido o QoS 1 para todas as mensagens enviadas e recebidas no protocolo MQTT, pois esta qualidade de serviço garante que a mensagem será entregue para o assinante, não foi necessária a utilização do QoS 2, pois durante a construção da aplicação foi realizado um tratamento no Node-Red para evitar ações duplicadas.

O diagrama de blocos da figura 6, possui uma representação completa do atuação

do sistema proposto, onde um novo visitante toca a campainha, essa informação é transmitida para o ESP32, o qual então realiza a publicação no tópico do broker, o dispositivo Android que está inscrito nesse tópico recebe a informação que existe uma pessoa tocando a campainha, então notifica o usuário através de uma mensagem e a vibração do aparelho. O usuário então acessa o aplicativo, que também possui a conexão com a câmera IP, podendo visualizar quem está em sua porta, então toma a decisão de abri-la ou não e publica essa informação em um tópico no broker, o ESP32 que está inscrito nesse tópico recebe a informação e então envia o comando para o relé atuar ou não na fechadura elétrica, permitindo ou não a entrada do visitante.

Figura 6 : Diagrama de Blocos do Sistema



Fonte: Autoria própria

### 3.1 Etapa 1 : Codificação do Aplicativo Android

Utilizando o software gratuito, disponibilizado pelo Google, Android Studio uma aplicação mobile foi criada na linguagem de programação Java, com o intuito de realizar a função monitoramento, controle e realizar autenticação de usuários.

A interface foi construída de forma a ser intuitiva para o usuário, permitindo um fácil acesso, e utilização.

A aplicação possui um sistema de cadastro, login, permitindo assim um controle e privacidade dos usuários, os dados de cada utilizador são salvos na Firebase do Google, um bando de dados em tempo real, hospedado na nuvem, permitindo assim atualizações constantes e cadastros de novos usuários sem interrompimento do aplicativo. Cada usuário passa por uma autenticação realizada também pela Firebase do Google, gerando uma chave única para cada pessoa utilizando a aplicação, evitando conflitos nos comandos, e garantido maior segurança nos dados.

No código da figura 7 é mostrado como é realizada a criação de um novo usuário no Google Firebase Store, após o novo utilizador inserir um nome e e-mail, esses dados são passados para um novo documento no formato do banco de dados, nesse código chamado de "documentodados", nesse documento são salvos o e-mail, e senha, bem como um número único de identificação de usuário

Figura 7 : Criação de um novo usuário

```
80     FirebaseFirestore db = FirebaseFirestore.getInstance();
81
82     Map<String, Object> usuarios = new HashMap<>();
83
84     usuarios.put("nome", nome);
85     usuarios.put("email", email);
86
87     id_usuario = FirebaseAuth.getInstance().getCurrentUser().getUid();
88
89     DocumentReference documentodados = db.collection("usuarios").document(id_usuario);
90
91     documentodados.set(usuarios).addOnCompleteListener(new OnCompleteListener<Void>() {
```

Fonte: Autoria própria

Na tela principal da aplicação está presente o código que realiza a conexão com o Broker MQTT, portanto assim que o usuário realiza o login no aplicativo, já é realizada a conexão com o Broker, evitando assim a necessidade do usuário ter que realizar manualmente conexões e desconexões com o Broker, que poderia acabar

se tornando uma tarefa cansativa e complexa.

Nesta tela também são realizados dois processos de extrema importância para a aplicação, por se tratar de uma automação destinada ao público com deficiência auditiva, é necessário que o celular vibre a cada pessoa nova que chegar na porta, conforme mostra o código na figura 8.

Figura 8 : Código Notificação e Vibração

```
132 @Override
133 public void messageArrived(String topic, MqttMessage message) throws Exception {
134
135
136     //vibração do celular
137     vibrator.vibrate( milliseconds: 500);
138
139     // Criar notificação push
140
141     NotificationCompat.Builder builder = new NotificationCompat
142         .Builder( context: this, MainActivity)
143         .setSmallIcon(R.drawable.logo)
144         .setContentTitle("Porta")
145         .setContentText("Uma pessoa está na porta")
146         .setPriority(NotificationCompat.FLAG_HIGH_PRIORITY);
147
```

Fonte: Autoria própria

No código apresentado, as duas funções somente são executadas quando uma nova mensagem é recebida no tópico MQTT que está inscrito, esse tópico é criado após o tratamento do Node-Red, e apenas apresenta novas mensagens quando uma pessoa aciona um botão, indicando que alguém está na porta.

Ao receber uma nova mensagem no tópico, as funções são executadas, a função vibrator é responsável por realizar a vibração do dispositivo Android, por 500 mili segundos, nesse trabalho não foi criado um padrão de vibração personalizado, apenas utilizado o disponível na própria função.

A segunda função, chamada de NotificationCompt.Builder, cria uma notificação para o usuário com o título Porta, e mostra que existe uma pessoa em sua porta, permitindo assim que sempre que um novo visitante chegar, o usuário

seja notificado.

Na tela de monitoramento são realizadas as principais funções da aplicação, possibilitando o usuário verificar por meio de um vídeo quem está em sua porta, e por meio de botões na tela, realizar a decisão de abrir ou não a mesma, ou pedir ajuda em casos de necessidade.

Todos estes comandos são enviados para o Mosquitto Broker, com suas respectivas ações, funcionando assim como ponte para comunicação, o Node-Red recebe essas ações e realiza seu tratamento filtrando as mensagens, que posteriormente serão enviadas para o ESP 32, o qual realiza as tarefas físicas designadas pelo usuário.

A análise do código de publicação do comando "abrir" é apresentada na Figura 9, e é importante ressaltar que o código está em conformidade com o modelo fornecido pela biblioteca Paho. Este código pode ser decomposto em vários componentes, conforme descrito a seguir:

String topico: Este é o tópico do Broker MQTT no qual a mensagem será publicada. Sendo o tópico é uma espécie de canal ou destino para a mensagem.

String mensagem: Refere-se à mensagem específica que será publicada no tópico. Esta mensagem é o comando que será transmitido para a ação desejada, que neste caso é "abrir".

client.publish: É a função utilizada para efetuar a publicação da mensagem no tópico selecionado. Esta função requer alguns parâmetros, incluindo o tópico ao qual a mensagem será enviada, a própria mensagem que será publicada e o nível de confiabilidade da mensagem.

É importante observar que todas as ações realizadas pelo usuário na tela de monitoramento seguem a mesma estrutura de código conforme descrito anteriormente, pois todas essas ações envolvem a publicação de mensagens no broker MQTT.

Figura 9 : Código para publicação no tópico

```
1 usage
216 public void publicar(View view) {
217
218     String topic = "AplicativoAndroid";
219     String message = "Abrirporta";
220
221     try {
222         client.publish(topic, message.getBytes(), qos: 0, retained: false);
223     }
224
225     catch (MqttException e) {
226
227         e.printStackTrace();
228     }
229
230 }
```

Fonte: Autoria própria

Como parte fundamental, as imagens da câmera IP são exibidas nessa tela, as imagens não são retidas na memória do aparelho celular, portanto para a transmissão e reprodução dessas informações foi escolhido um reproduutor de vídeo que suportasse o protocolo de transmissão RTSP, sendo necessário apenas a conexão de rede na câmera e no aparelho celular.

Na figura 10 é possível verificar como é realizada a construção do reproduutor de vídeo, conforme disponível na documentação do VLC Media Player para aplicativos mobile, na figura também é possível verificar que o IP da câmera é passado via URL, no objeto "media", e que as imagens possuem um buffer de 600 milissegundos, a escolha desse tempo de buffer foi realizada através de testes, e determinou-se que para a qualidade de imagem utilizada no projeto, esse tempo seria o necessário para evitar possíveis perdas na qualidade visual das imagens fornecidas.

Figura 10 : Código construção reprodutor de vídeo VLC

```
@Override
protected void onStart()
{
    super.onStart();

    mediaPlayer.attachViews(videoLayout, dm: null, subtitles: false, textureView: false);

    Media media = new Media(libVlc, Uri.parse(url));
    media.setHWDecoderEnabled(enabled: true, force: false);
    media.addOption(":network-caching=600");

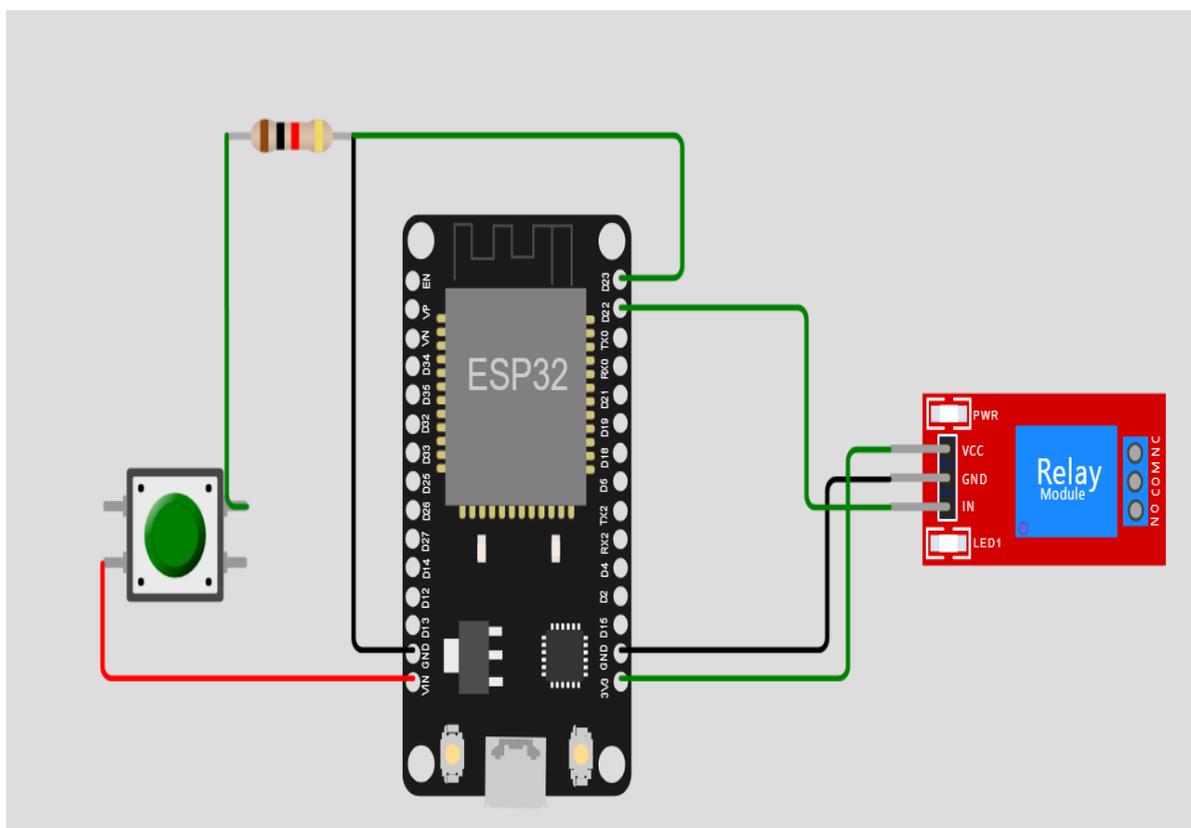
    mediaPlayer.setMedia(media);
    media.release();
    mediaPlayer.play();
}
```

Fonte: Autoria própria

### 3.2 Etapa 2 : Codificação da IDE Arduino e montagem do circuito

Durante esta fase do projeto, procedeu-se à elaboração de um código destinado à programação do microcontrolador ESP32, ao mesmo tempo em que se efetuou a montagem do circuito representado na figura 11. O mencionado circuito desempenha a seguinte funcionalidade: quando uma pessoa se aproxima da porta e aciona um botão, de forma análoga a um interfone, essa informação é então transmitida pelo ESP32 ao servidor de mensagens (Broker). Posteriormente, por meio de um dispositivo Android, é emitido um comando para controlar a ativação ou desativação do relé, resultando na abertura ou no fechamento da fechadura, conforme apropriado.

Figura 11 : Circuito de registro e acionamento



Fonte: Autoria própria

### 3.3 Etapa 3 : Criação do Flow no Node-Red

Durante essa etapa, foram realizados simultaneamente ajustes nos códigos do aplicativo, no código do ESP32 e foi montado um Flow no Node-Red, com o intuito de realizar a ponte de comunicação entre a aplicação móvel, e o dispositivo fixo.

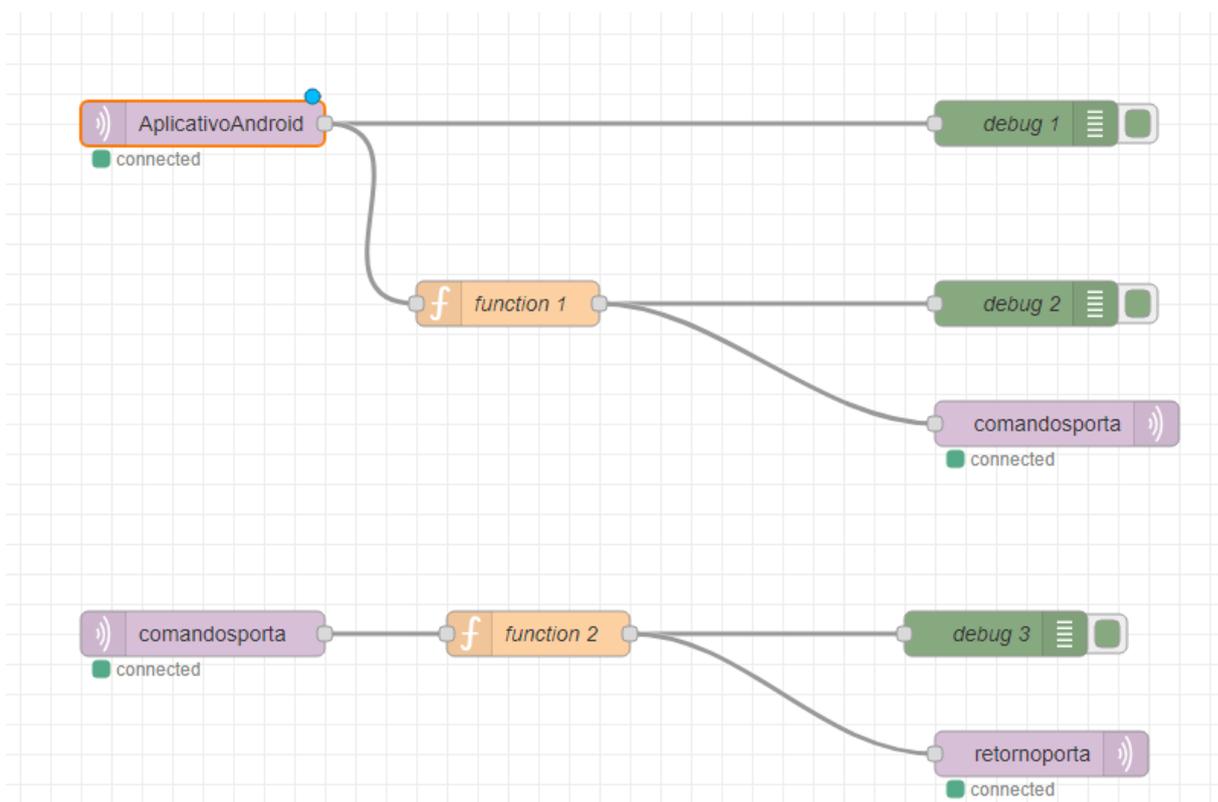
O Flow criado, possui o objetivo de atuar como um filtro das mensagens que são enviadas ao broker MQTT. Ao receber uma nova mensagem nos tópicos pré-definidos, é realizado um tratamento dos dados, após o tratamento os dados são enviados para seus respectivos tópicos.

Na figura 12, é possível observar o fluxo de dados, os quais são recebidos por meio do tópico "AplicativoAndroid". Essas informações passam por um processo de tratamento por uma função específica e, após o tratamento, são publicadas no tópico "comandosporta". Este último tópico é inscrito por dois clientes, o ESP32 e o Node-

Red, o que significa que tanto as publicações do Node-Red quanto as do ESP32 são processadas por uma segunda função. As mensagens resultantes desse processamento são então encaminhadas para o tópico "retornoporta," ao qual o aplicativo móvel está inscrito. Esse tópico desempenha um papel fundamental na ativação do sistema de notificações do dispositivo Android.

Além disso, com o intuito de controlar o fluxo de comandos e garantir a manutenção eficaz dos dados, foi implementada a função de "debug" em todas as saídas de informações. Essa função fornece uma representação visual dos dados sendo transmitidos, o que facilita a detecção e correção de eventuais erros no processo de transmissão. Essa abordagem contribui para aprimorar a confiabilidade e a eficiência do sistema, assegurando que a comunicação ocorra de maneira suave e precisa.

Figura 12 : Flow para tratamento de comandos MQTT



Fonte: Autoria própria

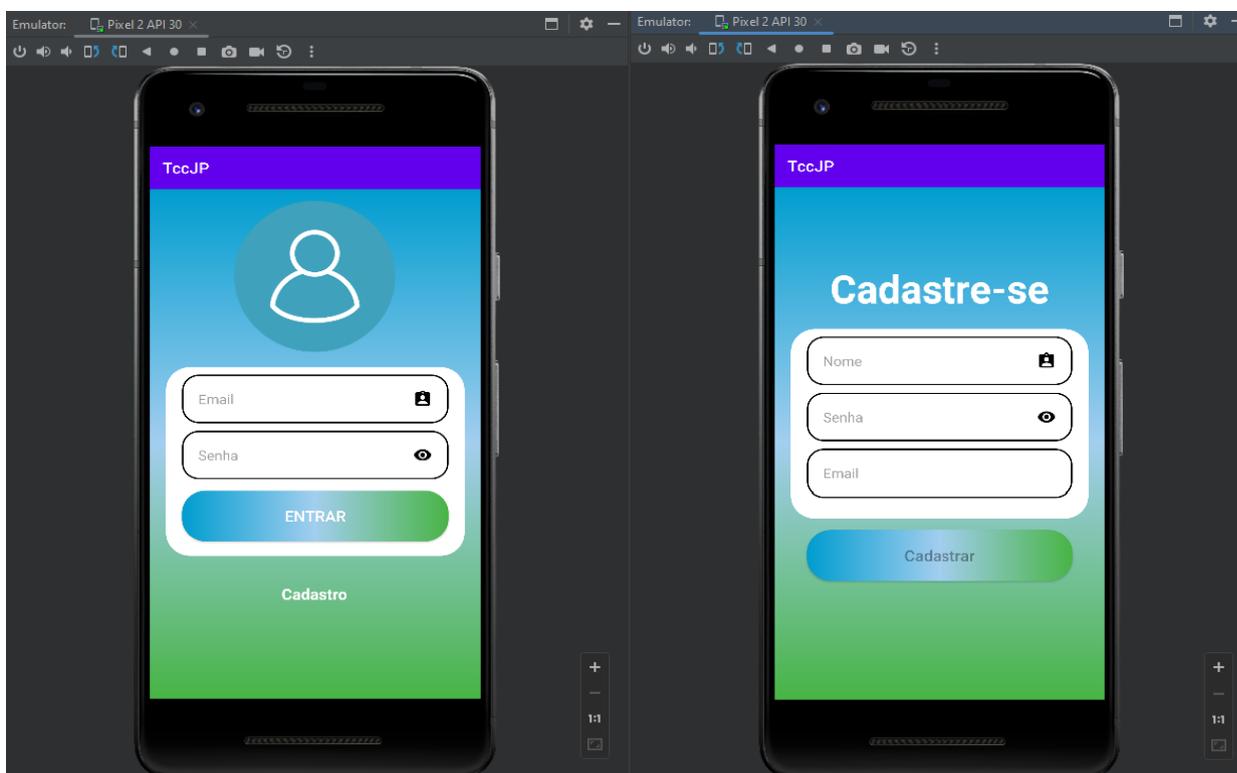
## 4 RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados os resultados provenientes da simulação de um dispositivo móvel utilizando o software Android Studio, bem como suas interações em um ambiente real. Importante frisar que, no ambiente real, optou-se por simular a ativação de uma fechadura eletrônica por meio de uma lâmpada LED 220V.

A escolha de utilizar o Android Studio para a simulação e não um dispositivo real, se deu em virtude do caráter expositivo do projeto, e utilizando um simulador torna-se mais ilustrativo e de melhor compreensão.

Conforme evidenciado na Figura 13, ao iniciar o aplicativo, o usuário é recepcionado por uma interface, proporcionando a opção de acesso mediante inserção de seu e-mail e senha, ou, caso não possua um cadastro ativo, realizar o processo de cadastro no aplicativo.

Figura 13 : Telas de Login e Cadastro do Aplicativo



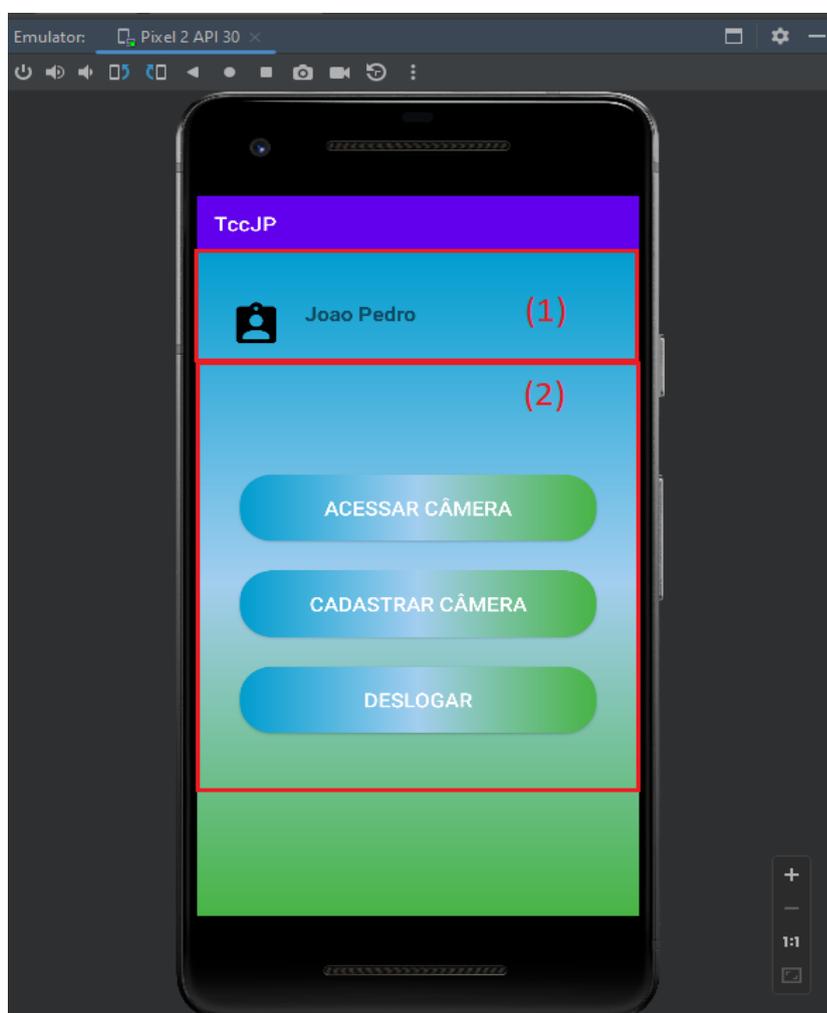
Fonte: Autoria própria

Após o usuário efetuar seu login no aplicativo, ele será direcionado para a tela

da figura 14. Nessa tela o usuário possui sua identificação (1), a qual foi fornecida na tela de cadastro, e um menu com algumas opções (2).

O botão Acessar Câmera leva o usuário até a sua tela de monitoramento que possui a função de fornecer a imagem da câmera de segurança do usuário bem como realizar a abertura de sua fechadura, o botão cadastrar câmera envia o usuário para uma tela onde ele pode inserir o IP de seu equipamento de segurança, permitindo assim uma maior liberdade para trocas de câmeras, ou adição de novos dispositivos de segurança. Por fim o botão de deslogar permite ao usuário finalizar sua sessão caso deseje, encerrar sua sessão.

Figura 14 : Tela Principal do Aplicativo

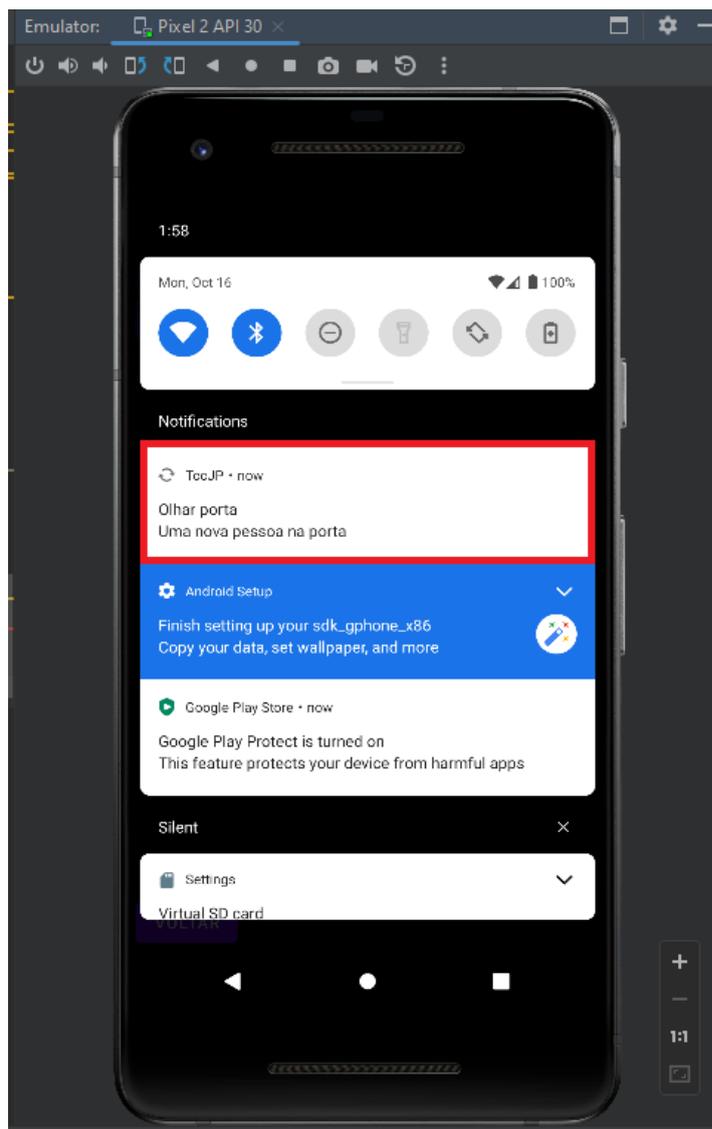


Fonte: Autoria própria

Conforme previamente detalhado na seção 3.1, o aplicativo apresenta a funcionalidade de notificação para informar o usuário sobre a chegada de novas

pessoas em sua residência. Essa notificação é exemplificada na figura 15, onde o usuário pode visualizar que um novo visitante que está chegando. Essa funcionalidade de notificação desempenha um papel fundamental na aplicação, ao fornecer informações em tempo real sobre as visitas à residência, de forma visual e sensorial , contribuindo para a segurança e independência do usuário.

Figura 15 : Notificação de novo visitante

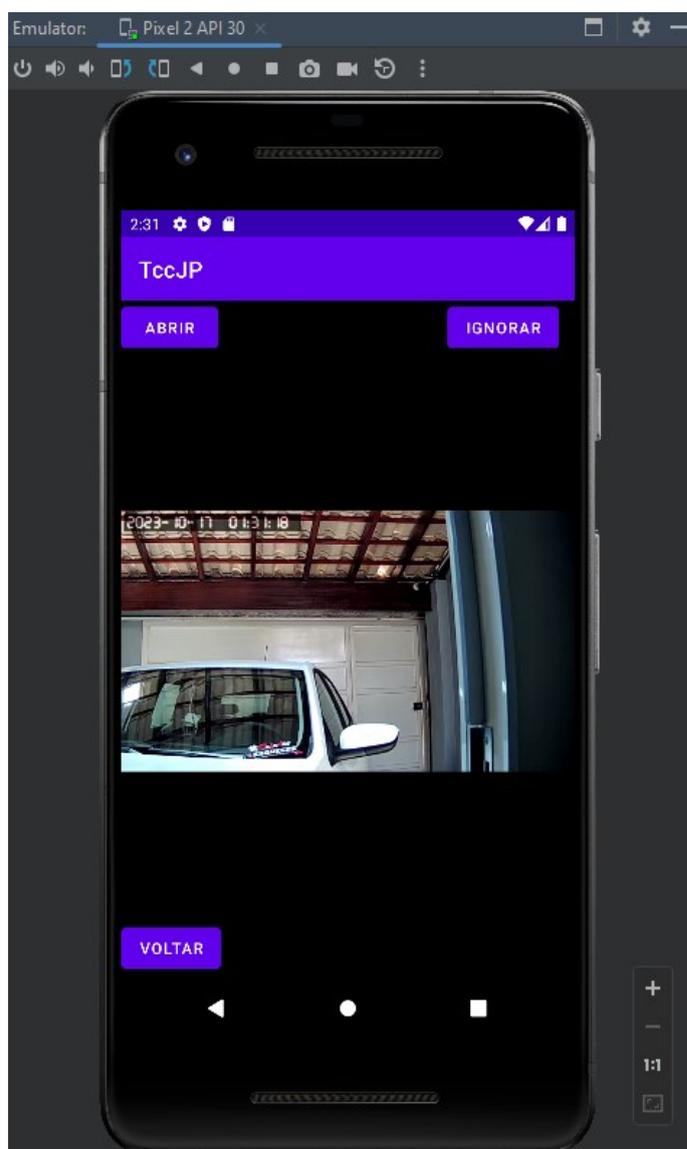


Fonte: Autoria própria

Na tela de monitoramento o usuário possui uma série de funções. Inicialmente ele possui a capacidade de visualizar as imagens geradas pela câmera IP, fornecendo uma visão em tempo real do ambiente, o usuário também possui a opção de controlar a abertura ou o fechamento do seu portão através do sistema de

trava eletrônica. Como anteriormente explicado na seção 3.1, essa comunicação ocorre mediante a utilização do protocolo MQTT, que garante uma comunicação eficiente e segura entre os dispositivos envolvidos. Dessa forma, o usuário tem o controle total sobre o acesso à sua propriedade, bem como a capacidade de monitorar visualmente a área em questão, proporcionando um ambiente de segurança e conveniência. Essa tela pode ser observada na figura 16.

Figura 16 : Tela de Monitoramento

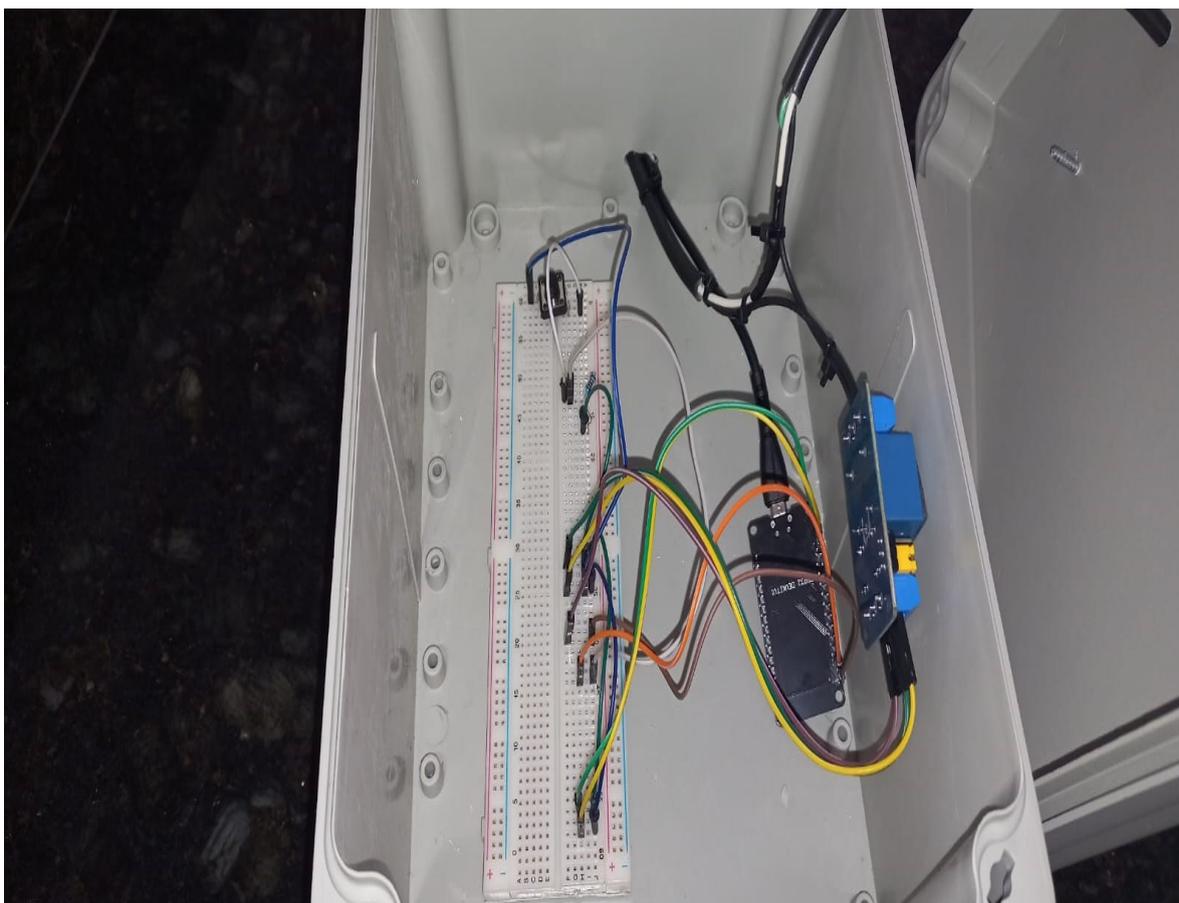


Fonte: Autoria própria

Conforme já mencionado anteriormente, foi desenvolvido um protótipo que tem como objetivo simular o funcionamento da fechadura e a interação do aplicativo

com essa fechadura. Uma representação desse circuito pode ser encontrado na seção 3.2 do documento, fornecendo uma representação visual das conexões e componentes envolvidos. Na figura 17, é possível observar o resultado da montagem desse protótipo.

Figura 17 : Circuito real com Esp32 e Relé



Fonte: Autoria própria

A fim de efetuar a simulação do processo de atuação sobre a fechadura, foi empregada uma lâmpada LED com uma tensão de 220V. Essa lâmpada é ativada sempre que o comando de abertura é enviado pelo aplicativo. O uso da lâmpada LED como indicador de ativação oferece uma demonstração visual do funcionamento do sistema, permitindo que os usuários observem de maneira direta quando o comando de abertura é executado com sucesso. Isso contribui para a compreensão prática do sistema de controle da fechadura.

O circuito em acionamento pode ser visualizado na figura 18, onde tem-se a fechadura fechada, ou sem acionamento, no lado esquerdo e a fechadura aberta, ou acionada, no lado direito.

Figura 18 : Circuito desligado e acionado



Fonte: Autoria própria

## 5 CONCLUSÃO

Este projeto permitiu a criação de um protótipo de um sistema de automação residencial projetado especialmente para pessoas com deficiência auditiva, com destaque para o uso do protocolo MQTT e ambiente de programação Android Studio. Ao longo desse trabalho, uma variedade de ferramentas e tecnologias foi aplicada, incluindo o Node-Red, Android Studio, Google Firebase, Mosquitto Broker e o microcontrolador ESP32.

Uma das principais constatações desse projeto foi a notável flexibilidade do protocolo MQTT, que se revelou eficaz na comunicação entre um microcontrolador e um dispositivo móvel, proporcionando uma comunicação rápida e de baixa latência. Além disso, ficou evidente a considerável capacidade de expansão desse protocolo, uma vez que, por meio de um broker, é possível estabelecer conexões entre dispositivos de fabricantes diversos. Nesse trabalho, essa característica foi exemplificada pela conexão bem-sucedida entre um microcontrolador ESP32 e um dispositivo móvel.

A pesquisa também incluiu um estudo aprofundado sobre automação residencial, com o intuito de identificar abordagens para concretizar a proposta inicial. Inicialmente, planejou-se utilizar dispositivos prontos disponíveis no mercado. Contudo, à medida que o projeto avançou e mais pesquisas foram realizadas na área, ficou claro que a melhor solução seria o desenvolvimento de um dispositivo próprio, essa mudança foi necessária visando uma busca pela redução de custos e pelo desejo de proporcionar uma experiência ao usuário mais fluída na utilização de seu sistema de automação.

O sistema desenvolvido buscou reduzir a quantidade de programação que o usuário necessita de realizar para utilizar o software, de modo a ser atrativo também para novas pessoas que sem grandes conhecimentos de tecnologias de automação, durante o projeto foi verificado que existe uma grande variedade de câmeras que não são totalmente compatíveis com outros componentes, essa falta de compatibilidade foi trabalhada no software do projeto, permitindo assim que o usuário não necessite realizar buscas de compatibilidade, apenas compre e utilize os componentes que mais lhe agradem.

Essa busca com aumento de compatibilidade busca aumentar a disseminação

da tecnologia entre pessoas que possuem um baixo contato com programação e noções de compatibilidade, sendo a única programação necessária do usuário a inserção do número de IP de sua câmera na aba de “Cadastro de Câmera” no aplicativo, e também oferecendo a livre escolha do usuário de trocar o número de IP da câmera quando desejar.

Durante a finalização do projeto foi constatada uma redução significativa dos custos do aparelho, em comparação com soluções do mercado, essa redução foi proporcionada graças a utilização de um microcontrolador e seus componentes básicos, o que proporcionou uma maior customização e possibilidade de alcançar resultados baratos e com qualidade próxima aos encontrados atualmente no mercado de automação residencial.

Além disso, essa pesquisa ofereceu oportunidades significativas para a aprimoração do conhecimento e das habilidades de programação de software, com ênfase na linguagem Java. Durante os estudos da linguagem Java, ficou evidente que o como é uma ferramenta versátil com ampla aplicação em diversos campos, incluindo IoT, desenvolvimento móvel, sistemas embarcados entre outros.

O trabalho alcançou as metas estabelecidas, que incluíam o desenvolvimento de um protótipo de campanha adaptada de baixo custo para atender às necessidades de pessoas com deficiência auditiva. No entanto, ao longo do projeto, foram identificadas algumas áreas que podem ser aprimoradas em futuros trabalhos.

Dentre as melhorias sugeridas está a conversão do código Java para a linguagem Flutter, o que pode resultar em melhorias significativas no desempenho do aplicativo e ampliar a compatibilidade com dispositivos mais antigos, aumentando o total de pessoas com acesso ao aplicativo e uma maior versatilidade na manutenção do código.

Outra melhoria proposta é a substituição do reprodutor de vídeo utilizado no aplicativo, por um que apresente suporte a outros protocolos de transmissão de vídeo além do RSTP, o qual foi utilizado nesse trabalho, essa mudança aumentaria a compatibilidade com diversas câmeras IPs presentes no mercado. Além disso, para reforçar a segurança, é recomendado um estudo sobre os protocolos de envio de mensagens em aplicativos de mensagens, visando a implementação de um botão de assistência que permita aos usuários solicitar ajuda de uma pessoa de confiança quando necessário.

## REFERÊNCIAS

CONCEIÇÃO, Wellington Nogueira Elizeu Da; COSTA, Romualdo Monteiro De Resende . Análise do Protocolo MQTT para Comunicação IoT através de um Cenário de Comunicação. 2023. Disponível em : <https://seer.uniacademia.edu.br/index.php/cesi/article/download/1688/1231>. Acesso em: 23 set. 2023.

CUNHA, Alessandro F. . O que são sistemas embarcados?, 2007. Disponível em: [https://files.comunidades.net/mutcom/ARTIGO\\_SIST\\_EMB.pdf](https://files.comunidades.net/mutcom/ARTIGO_SIST_EMB.pdf). Acesso em: 11 ago. 2023.

CURVELLO, Andre. Apresentando o módulo ESP8266. [S. l.], 2015. Disponível em: <https://embarcados.com.br/modulo-esp8266/>. Acesso em: 31 jan. 2023.

ESPRESSIF. ESP32. Espressif, 2023. Disponível em: <https://www.espressif.com/en/products/socs/esp32>. Acesso em: 20 set. 2023.

FREITAS, Diógenes Souza; SILVA , Luca Ananias Moraes. CAMPAINHA PARA PESSOAS COM DEFICIÊNCIA AUDITIVA. Docplayer, 2016. Disponível em: <https://docplayer.com.br/57847201-Campainha-para-pessoas-com-deficiencia-auditiva-diogenes-souza-freitas-1-luca-ananias-moraes-da-silva-2.html>. Acesso em: 25 ago. 2023.

HELLMANN, Rafael Alexandre Freiberger. Aplicativo Android e website interativos para busca de menores preços de produtos com código de barras. 2016. 90 f. TCC (Graduação) - Curso de Engenharia Elétrica, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2016. <https://seer.uniacademia.edu.br/index.php/cesi/article/download/1688/1231>. Acesso em: 23 set. 2023.

IBM . Dispositivos Suportados pelo IBM WebSphere MQ Telemetry. IBM MQ, 2021. Disponível em: <https://www.ibm.com/docs/pt-br/ibm-mq/7.5?topic=ssfksj-7-5-0-com-ibm-mq-pro-doc-q003060--htm>. Acesso em: 20 set. 2023.

MELLO, Marcio. O que é um microcontrolador. [S. l.], 25 abr. 2022. Disponível em: <https://victorvision.com.br/blog/o-que-e-um-microcontrolador/>. Acesso em: 1 fev. 2023.

MOREIRA, Paula Pfeifer . Quantos surdos há no Brasil e no mundo em 2023? A VERDADE. Crônicas da Surdez, 2023. Disponível em: <https://cronicasdadasurdez.com/quantos-surdos-no-mundo/>. Acesso em: 27 fev. 2023.

NERI, Renan *et al.* MQTT. GTA URFJ, 2019. Disponível em: <https://www.gta.urfj.br/ensino/eel878/redes1-2019-1/vf/mqtt/>. Acesso em: 26 set. 2023.

OLIVEIRA, Ricardo Rodrigues. USO DO MICROCONTROLADOR ESP8266 PARA AUTOMAÇÃO RESIDENCIAL. pantheon ufr, 2017. Disponível em: <https://pantheon.ufrj.br/bitstream/11422/18188/1/monopoli10019583.pdf>. Acesso em: 05 set. 2023.

ORGANIZATION, World Health. World report on hearing. [S. l.], 2022. Disponível em: <https://www.who.int/publications/i/item/9789240020481>. Acesso em: 27 fev. 2023.

REIS, Pedro Guilherme Pereira Dos. GEOFENCING APLICADO À COMERCIALIZAÇÃO DE PRODUTOS ELETRÔNICOS EM PLATAFORMA MOBILE. repositorio puc goias, 2023. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/bitstream/123456789/2855/1/TCC%20-%20Pedro.pdf>. Acesso em: 30 set. 2023.

SILVA JÚNIOR, Esdras Barbosa Lima . Node-RED KNoT: Um módulo de integração da ferramenta Node-RED com a meta plataforma KNoT. Repositorio UFPE, 2020. Disponível em: [https://www.cin.ufpe.br/~tg/2020-3/TG\\_EC/tg\\_eblsj.pdf](https://www.cin.ufpe.br/~tg/2020-3/TG_EC/tg_eblsj.pdf). Acesso em: 07 mar. 2023.

SILVA, Adrieri Marrie Costa et al. DESENVOLVIMENTO DE PROJETO PARA PROTÓTIPO DE CAMPAINHA ADAPTADA PARA SURDOS. Repositorio IFBA, 2022. Disponível em: [http://repositorio.ifba.edu.br/jspui/bitstream/123456789/333/4/TCC\\_Equipe%20Adri\\_2022.pdf](http://repositorio.ifba.edu.br/jspui/bitstream/123456789/333/4/TCC_Equipe%20Adri_2022.pdf). Acesso em: 29 ago. 2023

SOUZA, Fabio Cassio De et al. IIOT UTILIZANDO PROTOCOLO MQTT. repositorio animaeducacao, 2023. Disponível em: <https://repositorio.animaeducacao.com.br/bitstream/ANIMA/18962/1/IIOT%20UTILIZANDO%20PROTOCOLO%20MQTT.pdf>. Acesso em: 20 set. 2023.

ZANATTA, Bianca . Casa 'inteligente' é cada vez mais realidade. Aureside, 2021. Disponível em: [http://www.aureside.org.br/\\_pdf/oesp\\_170121.pdf](http://www.aureside.org.br/_pdf/oesp_170121.pdf). Acesso em: 05 jan. 2023.