

INSTITUTO FEDERAL DE SANTA CATARINA

ANDREY GONÇALVES

**Sistema de monitoramento e alertas  
emergenciais para idosos**

São José - SC

agosto/2022

# **SISTEMA DE MONITORAMENTO E ALERTAS EMERGENCIAIS PARA IDOSOS**

Monografia apresentada ao Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheiro de Telecomunicações.

Orientador: Mario de Noronha Neto

Coorientadora: Professor Coorientador da Silva

São José - SC

agosto/2022

# RESUMO

Com o envelhecimento da população surgiram lacunas no cuidado e monitoramento de idosos. Com o avanço tecnológico viabilizou a criação de sistemas para monitoramento. Tendo em vista isso, a proposta do projeto é apresentar uma solução para auxiliar o monitoramento de pessoas idosas e seu ambiente domiciliar. A ideia principal consiste no desenvolvimento de um *gateway* Message Queuing Telemetry Transport (MQTT) utilizando uma Raspberry Pi 4 com um módulo de comunicação Zigbee para processar os sinais de diversos sensores Zigbee de mercado e uma *stack* de um cliente Session Initiation Protocol (SIP) para comunicação de voz em casos emergenciais ou para assistência ao contratante. As informações de monitoramento do idoso e do ambiente será exibida em um *Dashboard* na plataforma TagoIO com uma interface amigável, onde o encarregado pelo monitoramento pode verificar em tempo real os dados coletados no ambiente de instalação. Vale ressaltar que os dados coletados para desenvolvimento do sistema são de sensores de temperatura, umidade, fumaça, gás de cozinha, porta, botão de emergência, vibração para detecção de quedas, vazamento de água e presença. Além disso o produto tem duas botoeiras onde pode ser acionada para ligar para números pre-definidos, como uma central de atendimento de saúde ou algum familiar. O produto possui alto falante e microfone embutido e é capaz de originar e receber chamadas SIP.

**Palavras-chave:** Telecomunicador SIP. Monitoramento de idosos. Sensores Zigbee.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>4</b>
1.1	Objetivo geral	5
1.2	Objetivos específicos	5
1.3	Estrutura do trabalho	5
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>6</b>
2.1	ZigBee	6
2.2	MQTT	8
2.3	Zigbee2MQTT	10
2.4	SIP	11
2.5	TagIO	13
<b>3</b>	<b>APRESENTAÇÃO DA PROPOSTA</b>	<b>14</b>
<b>3.1</b>	<b>Descrição do sistema</b>	<b>14</b>
3.1.1	Descrição dos componentes	15
3.1.1.1	Raspberry Pi 4	15
3.1.1.2	SONOFF ZigBee 3.0	17
3.1.1.3	Alto falante e microfone	18
3.1.1.4	Sensores Radiofrequência (RF) <i>ZigBee</i>	18
3.1.1.5	Cliente SIP	18
<b>3.2</b>	<b>Desenvolvimento do sistema</b>	<b>19</b>
3.2.1	Desenvolvimento do gateway	19
3.2.1.1	Instalação do sistema operacional	19
3.2.1.2	Instalação do softphone Linphone	19
3.2.1.3	Monitoração da GPIO - botoeira	19
3.2.1.4	Instalação Zigbee2MQTT	19
3.2.2	Desenvolvimento da interface	19
3.2.2.1	Interface de configuração do produto	19
3.2.2.2	Interface de monitoração - Dashbord	20
3.2.3	desenvolvimento do protótipo	20
<b>3.3</b>	<b>Cronograma de execução</b>	<b>20</b>
<b>4</b>	<b>CONCLUSÕES</b>	<b>21</b>
	<b>REFERÊNCIAS</b>	<b>22</b>

# 1 INTRODUÇÃO

O envelhecimento da população está acontecendo mais rápido do que nunca. Segundo o relatório mundial de envelhecimento e saúde da [Organização Mundial da Saúde \(OMS\)](#) ([OMS, 2015](#)), em 2050 o número de pessoas com idade superior a 60 anos será o dobro que em 2015, esta fatia será cerca de um quinto da população do mundo, por volta de 2 bilhões idosos.

Uma grande proporção da população idosa sofre de problemas de saúde relacionado à idade, como a doença de Alzheimer, demência, doença cardiovascular, diabetes ou outras doenças crônicas. Além disso temos o declínio nas habilidades físicas e cognitivas dos idosos onde isso resulta em quedas. Mas com o avanço tecnológico é possível criar sistemas para monitoramento e comunicação com os idosos em sua residência.

O mercado brasileiro oferece alguns sistemas e serviços de cuidados de idosos e pessoas debilitadas, porém são soluções com tecnologias antigas, baseadas em linhas de telefonia analógica denominada [Public Switched Telephone Network \(PSTN\)](#), telefonia [Sistema Global para Comunicações Móveis \(GSM\)](#) ou [Quarta Geração de telefonia móvel \(4G\)](#). A solução proposta nesse artigo é baseado em tecnologia digital, com integração completa pela internet, o que traz ao produto maior confiabilidade e velocidade para a tomada de decisões. Agregando estas novas tecnologias, também é possível ter uma gama de cuidados e prevenções muito maiores. O produto será desenvolvido com a concepção de uso para uma plataforma inovadora social de idosos, agregando sensores para detecção de riscos e rápido atendimento. O produto possuirá um intercomunicador IP que permite enviar ligações para uma central de atendimento ou algum contato pré determinado. Através dos sensores, é monitorado possíveis situações de risco no ambiente ou com o usuário do sistema e estes dados são enviados via rádio comunicação ([Zigbee](#)) para o intercomunicador que imediatamente reporta como um alarme para a central. Os sensores utilizados na proposta desta monografia para monitorar o ambiente são: botão de emergência (colar/pulseira), sensor que detecta queda, sensor de gás de cozinha, sensor fumaça, sensor de temperatura e humidade, e sensor de porta. Um *dashboard* de monitoramento para verificar os dados coletados pelos sensores será desenvolvido utilizando a plataforma [TagoIO](#) Além disso o produto será um comunicador [SIP](#) com alto falante e microfone embutido, onde é possível originar uma chamada de emergência ou até mesmo requisitar algum tipo de atendimento com uma central de monitoramento de saúde. Desta forma a solução oferece um conjunto completo de serviços e soluções de monitoramento remoto para ajudar os idosos a manter uma vida saudável e segura. Podendo ter serviços de tele assistência projetados especificamente para ajudar os idosos a manter a independência

e viver com segurança em sua casa.

## 1.1 Objetivo geral

Desenvolver um sistema de monitoramento e alertas emergenciais para idosos utilizando a tecnologia *Zigbee* para comunicação RF e os protocolos MQTT e SIP para a comunicação de dados entre o *gateway* e a aplicação.

## 1.2 Objetivos específicos

- Analisar a viabilidade técnica para o desenvolvimento do sistema
- Entender o funcionamento das tecnologias *Zigbee* e dos protocolos MQTT e SIP
- Especificar a plataforma de desenvolvimento utilizada e adquirir os componentes do sistema
- Desenvolver e testar isoladamente os componentes do sistema
- Desenvolver o *front-end* do sistema integrado
- Integrar os componentes do sistema e realizar testes com o sistema integrado

## 1.3 Estrutura do trabalho

O artigo está estruturado da seguinte forma: No [Capítulo 1](#) do documento encontra-se a introdução e o objetivo a ser atingido. Em seguida, no [Capítulo 2](#) é reservada para fundamentação teórica onde será exposto as tecnologias a serem utilizadas e os produtos existentes no mercado. No [Capítulo 3](#) é exibido a apresentação da proposta desenvolvida descrevendo os componente utilizados para criação do produto, as atividades desenvolvidas e o cronograma de execução do projeto. E finalmente no [Capítulo 4](#) a conclusão da monografia.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 ZigBee

Conforme aponta Stoll (2008), o *ZigBee* é uma normal técnica, baseado no padrão *IEEE 802.15.4*, para comunicações sem fio de baixa potência e curta distância. O protocolo *ZigBee* foi desenvolvido pelo grupo *ZigBee Alliance*, uma associação sem fins lucrativos com o objetivo de “construir uma norma aberta com aplicações que permitam interoperabilidade entre diversos fabricantes, apoiar a base para certificação de produtos compatíveis com *ZigBee* e promover a tecnologia” (STOLL, 2008).

Podemos encontrar o protocolo *ZigBee* em diversas aplicações, sendo algumas delas relacionadas com automação e controle predial, controle industrial, periféricos para PC, controle remoto de produtos eletrônicos, automação residencial e comercial e, saúde pessoal, conforme pode ser visualizado na Figura 1 (ANDRIGHETTO, 2008).

Figura 1 – Aplicações para o *ZigBee*.



Fonte: ROGERCOM, 2010 apud (KINDERMANN; SANTOS, 2012)

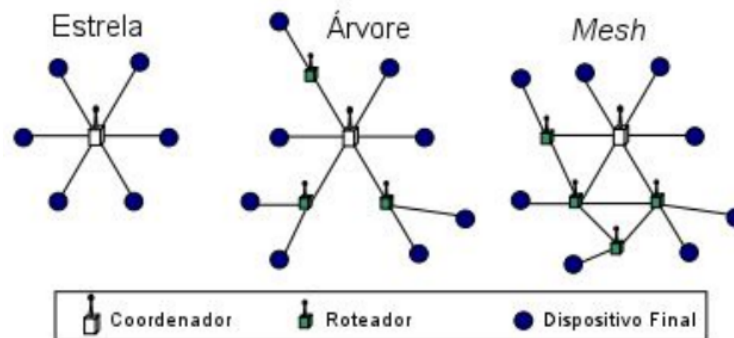
O padrão *IEEE* define para as redes *ZigBee* dois tipos de dispositivos, sendo eles classificados como *RFD* (*Reduced Function Device*) ou *FFD* (*Full Function Device*). A primeira classificação é referente aos dispositivos que podem funcionar apenas como *end-pointings* da rede, pois podem apenas ter comunicação com dispositivos *FFD*, sendo assim, são dispositivos mais simples e de menor custo, pois visam um consumo de energia reduzido. Os dispositivos *FFD*, por sua vez, estão aptos a funcionar em qualquer um dos modos de configuração padrão da rede *ZigBee* (coordenador, roteador ou dispositivo final), exigindo um hardware mais potente, pois são configurados para utilizar todos os recursos da pilha protocolar, tendo um consumo maior de energia (KINDERMANN; SANTOS, 2012).

Sendo assim, os dispositivos que pertencem a uma rede *ZigBee* são classificados de acordo com a configuração padrão a que pertencem, sendo elas conforme a seguir:

- Dispositivo final *ZigBee* (DFZ) - atua como um simples dispositivo na rede;
- Roteador *ZigBee* (RZ) - possui as funcionalidades de roteamento da rede;
- Coordenador *ZigBee* (CZ) - dispositivo que gerencia a rede em que está inserido.

Além disso, a rede *ZigBee* suporta as topologias de rede em estrela, árvore e em malha, conforme pode ser observado na Figura 2 (SANTOS, 2015).

Figura 2 – Topologias de redes *ZigBee*.



Fonte: BARONTI, 2007 apud (SANTOS, 2015).

A topologia de rede em estrela é projetada com foco na comunicação de um nó para vários nós. A topologia em árvore utiliza de mecanismos hierárquicos de árvore de roteamento e, as redes *mesh* utilizam um método de encaminhamento misto, tendo como uma das principais qualidades a redundância na rede, fazendo com que caso a comunicação entre alguns dispositivos falhe, o roteamento dos dados na rede não será afetado (SANTOS, 2015).

As aplicações *ZigBee* diferem das *Wi-Fi* e *Bluetooth Classic* não apenas em relação à topologia permitir a configuração de redes *mesh*, mas também em relação às implementações com distâncias mais curtas e custos por dispositivos mais baixos. O protocolo foi desenvolvido para utilização em sistemas embarcados que necessitem de curto alcance de comunicação, trabalhando com faixas de distância entre 30m e 100m para ambientes internos e atingindo até 1500m em áreas abertas (KINDERMANN; SANTOS, 2012).

Kindermann e Santos (2012) trazem que o protocolo *ZigBee* opera nas faixas de frequência *ISM* (*Industrial, Scientific and Medical*), sendo elas:

- 868 MHz na Europa;
- 915 MHz nos Estados Unidos;



- 2,4 GHz para o resto do mundo.

Além disso, o padrão *ZigBee* não necessita de licenciamento para seu funcionamento, permitindo que a tecnologia seja usada livremente e, devido suas características de rede e criptografia de dados *AES* (*Advanced Encryption Standard*) de 128 bits, oferece uma notável imunidade a ruídos, minimizando as chances de interferência entre os dispositivos configurados em redes distintas ou que utilizem da mesma faixa de frequência (KINDERMANN; SANTOS, 2012).

## 2.2 MQTT

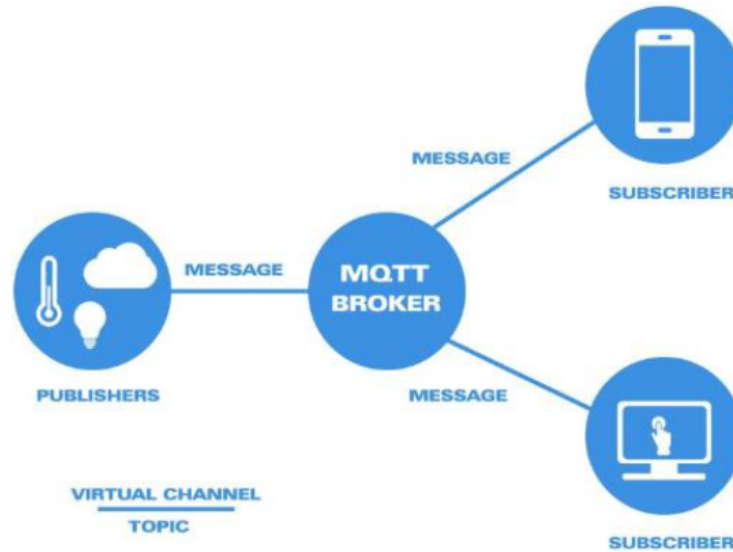
Martins e Zem (2015, p. 72) conceituam que:

“na Internet ‘tradicional’ existem diversos protocolos de comunicação responsáveis por gerenciar a transferência de dados em uma rede que conecta dois ou mais computadores. Para citar alguns exemplos: HTTP, FTP e SFTP. Quando se fala em comunicação entre dois ou mais dispositivos (ou um conjunto de aplicações) conectados em uma rede também surge a necessidade de pensar em um protocolo que gerencie esta comunicação, isto é, a troca de mensagens e/ou dados entre os elementos que compõem esta rede de objetos de forma eficiente considerando as características e limitações impostas pelo ambiente” (MARTINS; ZEM, 2015, p. 72).

Dessa forma, surgem protocolos de transferência que podem ser utilizados em ambientes restritos, sendo um deles, o protocolo *MQTT* (*Messaging Queue Telemetry Transport*). O *MQTT* foi criado em meados de 1999 e trata-se de um protocolo voltado para dispositivos restritos e redes inseguras, com baixa largura de banda e alta latência, sendo baseado na arquitetura *publish/subscribe* (MARTINS; ZEM, 2015).

Na arquitetura *publish/subscribe*, a qual o protocolo *MQTT* é baseado, o dispositivo é responsável por enviar as informações ao servidor - também chamado como *broker* -, que opera como intermediário retransmitindo as informações recebidas aos clientes interessados (CONCEIÇÃO; COSTA, 2019). A Figura 3, ilustra o modelo de comunicação utilizado pelo protocolo *MQTT*.

Figura 3 – Modelo de comunicação MQTT.



Fonte: (VALENZUELA et al., 2021).

Situado na camada de aplicação da arquitetura *TCP/IP*, o protocolo de comunicação MQTT pode ser aplicado sobre qualquer tecnologia, possibilitando a interconexão de dados e interações em tempo real entre sistemas, softwares e máquinas e, devido à isso desempenham um papel de alta importância em aplicações voltadas para projetos de Internet das Coisas (Internet of Things (IoT)) (VALENZUELA et al., 2021)).

Como principais características, que permitem uma boa compatibilidade do protocolo nas aplicações IoT, Valenzuela et al. (2021, p. 43410) elenca:

- “- Simples configuração: é possível fazer configurações com linhas de comando de baixa complexidade, sem a perda de padronização.
- Comunicação bidirecional: qualquer integrante de uma rede com MQTT pode desempenhar o papel de receptor e/ou de fornecedor de informação.
- Modelo publisher/subscriber: as informações são enviadas na estrutura de tópicos, assim o receptor não recebe informações que não sejam úteis para seu funcionamento.
- Assíncrono: o sistema utilizado permite um funcionamento em que o fornecedor de informações não necessita de confirmações para funcionar.
- Segurança: o protocolo MQTT permite a utilização de ferramentas para a segurança da rede, como por exemplo designar dados de login e senha para cada cliente conectado à rede. Em adicional, os brokers que são parte do protocolo podem utilizar ferramentas de criptografia.
- Baixo consumo de energia [...]
- Bom desempenho em áreas com problemas de transmissão.”

As mensagens de comando MQTT, por sua vez, possuem um cabeçalho fixo composto de dois bytes, onde o primeiro byte identifica o tipo da mensagem e campos marcadores como entrega duplicada (*DUP*), qualidade de serviço (*QoS*) e *RETAIN* (MARTINS; ZEM, 2015). A estrutura do cabeçalho fixo das mensagens do protocolo pode ser visualizada na Figura 4.

Figura 4 – Estrutura do cabeçalho fixo da mensagem MQTT.

bit	7	6	5	4	3	2	1	0
Byte 1	Tipo da Mensagem				Flag DUP	Nível QoS		RETAIN
Byte 2	Largura restante							

Fonte: (MARTINS; ZEM, 2015).

O cabeçalho da mensagem MQTT é composto por 7 bits. Em relação do primeiro byte, do bit 7 ao 4 é representado o tipo da mensagem. Os quatro bits restantes do primeiro byte dividem-se em quatro campos que indicam as preferências definidas antes do envio da mensagem, sendo:

- *Duplicate delivery (DUP)*: ocupa o bit 3 e possui como função indicar a entrega duplicada quando o cliente ou servidor tentam enviar mensagens que tenham *Quality of Service (QoS)* maior que 0 e requeiram *acknowledgment (ACK)*;
- *Quality of Service (QoS)*: ocupa os bits 1 e 2 e indica o nível de garantia de entrega de uma mensagem;
- *RETAIN*: ocupa o bit 0 e possui como função a retenção da mensagem de tipo *PUBLISH* no servidor mesmo depois de ser entregue aos clientes, quando o marcador estiver ativado. No evento de uma nova subscrição a um tópico, a última mensagem retida deve ser enviada para um novo cliente, caso o marcador esteja ativo (MARTINS; ZEM, 2015).

O protocolo MQTT encontra-se atualmente na versão 5.0 e conforme apresentado pelo (IBM, 2021) possui os recursos a seguir:

- Padrão Oasis;
- Aprimoramento para escalabilidade;
- Relatório de erro melhorado;
- Formulação de padrões comuns que incluem descoberta de recurso e resposta de solicitação;
- Mecanismos de extensibilidade que incluem propriedades do usuário;
- Melhorias de desempenho e suporte para clientes pequenos.

## 2.3 Zigbee2MQTT

O *ZigBee2MQTT* é um software de código aberto, desenvolvido por *Koen Kanters*, que provê uma interface para o usuário interagir com sistemas *ZigBee*. A função do

software é que se possa controlar múltiplos dispositivos *ZigBee* de diferentes fabricantes, através de MQTT (RUST, 2022).

## 2.4 SIP

Conforme definição do *RFC3261*, o protocolo SIP (*Session Initiation Protocol*) é um protocolo de controle da camada de aplicação que pode estabelecer, modificar e encerrar sessões multimídias pela Internet (ROSENBERG et al., 2002).

O SIP é um protocolo que ganha força perante aos desenvolvedores, devido à rapidez que as aplicações podem ser desenvolvidas, estando disponível mais cedo aos clientes. Trabalhando em conjunto com outros protocolos e de forma separada da mídia, o SIP é visto como o futuro das aplicações Voz sobre Protocolo de Internet (VoIP) (MARTINEZ; JR, 2011).

O protocolo SIP atua na camada de aplicação do modelo *OSI* e oferece cinco tipos de serviços para iniciação e finalização de sessões multimídias, que são descritas abaixo, conforme apresentado por Andreis (2010, p. 16):

- “- Localização do usuário: responsável pela localização do terminal para estabelecer a conexão
- Disponibilidade do usuário: responsável por realizar a vontade do usuário em estabelecer uma sessão de comunicação.
- Recursos do usuário: responsável pela determinação dos meios de comunicação e os parâmetros a serem utilizados.
- Gerência da sessão: responsável por inicializar, terminar ou colocar em espera uma sessão.
- Modificar sessão: responsável por modificar uma sessão em andamento.”

O SIP utiliza do protocolo *Session Description Protocol* (SDP) para estabilizar uma sessão multimídia. Este protocolo não entrega a mídia em si, mas é usado para a negociação entre os pontos finais de todas as propriedades associadas com o tipo de mídia (ANDREIS, 2010).

O transporte da mídia, por sua vez, é realizado pelo protocolo *Real-time Transport Protocol* (RTP), que define um formato de pacote padrão para a entrega de mídias através da Internet (ANDREIS, 2010).

Para o bom funcionamento do protocolo SIP, são definidas diversas entidades que juntas compõem a arquitetura e desempenham os seguintes papéis dentro do protocolo SIP, de acordo com Andreis (2010):

- User Agents: é a entidade que interage com o usuário, tendo a capacidade de enviar e receber requisições e dessa forma agindo tanto como cliente como servidor.

- Proxy Server: é uma entidade intermediária com a finalidade de fazer pedidos em nome de outros clientes, pode atuar tanto como servidor quanto cliente.
- Redirect Server: é um tipo de servidor SIP que ajuda a localizar entidades User Agents e fornece localizações alternativas, onde o usuário pode ser acessado.
- Register Server: é um tipo de servidor que armazena registros sobre usuários, fornecendo serviços de localização.

O protocolo SIP é baseado em texto e utiliza em sua estrutura o conjunto de caracteres *UTF-8*. Trabalha através de requisições e respostas para suas transações e suas respostas possuem um *status code*, que informa o resultado da requisição pela qual está respondendo. Conforme apresentado por [Martinez e Jr \(2011, p. 21\)](#), o *status code* varia de 100 a 699 e cada centena representa um tipo de resposta, tendo as seguintes classificações:

“1XX - Mensagem informativa  
 2XX - Mensagem de sucesso  
 3XX - Mensagem de redirecionamento  
 4XX - Mensagem de erro na requisição  
 5XX - Mensagem de erro do servidor  
 6XX - mensagem de erro global “

O *status code* ainda vem acompanhado de uma *reason phrase*, que é referente à uma frase informativa ao código da mensagem. O protocolo SIP possui diversos tipos de requisições, a maior parte delas integram a parte estendida do protocolo e não se tornam importantes para o funcionamento básico. Sendo assim, as principais requisições do SIP são de 6 tipos, conforme abaixo:

“INVITE - usado para convidar um usuário para uma sessão  
 ACK - usado para confirmar o recebimento de uma resposta final  
 OPTIONS - usado para perguntar ao servidor a respeito de suas funcionalidades  
 BYE - usado para sair de uma sessão  
 CANCEL - usado para cancelar transações pendentes  
 REGISTER - usado para que o usuário informe sua localização atual para um servidor de registro.” ([MARTINEZ; JR, 2011, p. 22](#))

Em relação à segurança, o SIP utiliza de diversos mecanismos adequados a diferentes aspectos e aplicações que proporcionem a autenticação de utilizadores e a privacidade dos participantes em uma sessão. Para a autenticação da identidade dos utilizadores, o protocolo SIP utiliza do método de autenticação *Digest*, utilizado pelo protocolo [Hypertext Transfer Protocol \(HTTP\)](#), permitindo aos utilizadores a identificação através de um nome de utilizador e de uma palavra-chave cifrada, no entanto, esse método não garante a confidencialidade e integridade das mensagens ([SOUSA; CARRAPATOSO, 2003](#)).

## 2.5 TagoIO

A **TagoIO** é uma plataforma web, de nuvem **IoT** de ponta a ponta. A finalidade da plataforma é permitir o monitoramento de ambientes via dispositivos **IoT** conectados à rede e é indicada para quem precisa de uma solução rápida e de fácil configuração para iniciar o sistema de monitoramento (XAVIER, 2019).

É possível a utilização em diferentes aplicações, tais como na agricultura, cadeia de mantimentos, refrigeração, automação industrial, logística, energia, entre outros. A plataforma permite que haja o gerenciamento de dispositivos, armazenamento de dados, execução de análises, integração de serviços e suporte e gerenciamento de usuários (TAGOIO, 2022).

O sistema permite que haja conexão com dispositivos **MQTT**, **HTTPS**, **WiFi**, **LoRaWAN**, **Sigfox**, **NB-IoT**, **LTE**, **BLE**, **ZigBee** e muitos outros, além de oferecer uma vasta gama de *widgets* para construção da *dashboard* (TAGOIO, 2022).

O primeiro passo do processo de criação da solução **IoT** consiste na conexão com um dispositivo que realize a leitura dos ambientes e envio dos dados para a plataforma. Tendo a conexão feita, o próximo passo é a escolha do *template* para iniciar a construção da aplicação e integração das soluções, seguindo para a criação e gerenciamento dos usuários e dispositivos, escalando a aplicação de forma rápida. Por fim, tendo a configuração feita, o *dashboard* no sistema estará pronto para o monitoramento e será possível criar diversos painéis por meio dos recursos e *widgets* disponibilizados pela plataforma (XAVIER, 2019).

A plataforma também oferta para os desenvolvedores espaço para criar *scripts* que manipulam dados de qualquer dispositivo em tempo real. Então podemos fazer uma infinitividade de manipulações, para geração de relatórios e implementações de aprendizado de máquina. Para tomada de decisões é possível enviar *SMS*, *e-mails*, notificações e dados de volta para o seu dispositivo, ou até mesmo executar os *scripts* previamente desenvolvido (TAGOIO, 2022).

# 3 APRESENTAÇÃO DA PROPOSTA

A proposta deste projeto consiste em desenvolver um *gateway Zigbee* para processar dados de diferentes sensores RF, também embarcar uma intercomunicação SIP para chamadas de voz. Na atual seção será especificado as etapas e requisitos do projeto.

## 3.1 Descrição do sistema

O sistema visa oferecer uma solução integrada para monitoramento de idoso e seu lar, através de sensores *ZigBee* pareados ao *gateway*, será possível receber em tempo real em um *dashbord* de fácil visualização eventos tal como de queda do idoso(atraves de um colar com sensor de vibração), informações do ambiente como temperatura, umidade, sensor de porta, sensor de gás de cozinha, sensor de fumaça e também um botão de emergência. Ao ser gerado um evento de emergência imediatamente o *gateway* irá realizar uma chamada SIP para uma central de monitoramento ou números previamente configurados.

O junto com o produto desenvolvido poderemos oferecer um serviço de tele assistência. Onde a central pode ligar para garantir que o idoso tome os devidos remédios corretamente, ou até mesmo para instigar exercícios no dia a dia.

Para a configuração do produto será desenvolvida uma interface de acesso à web de configuração, com usuário e senha. Um menu principal deixando a disposição submenus para certas configurações do dispositivo será entregue. Então poderemos realizar configurações de rede, configuração do ramal SIP, configurações de volume do áudio da chamada e também uma tela para pareamentos de novos sensores.

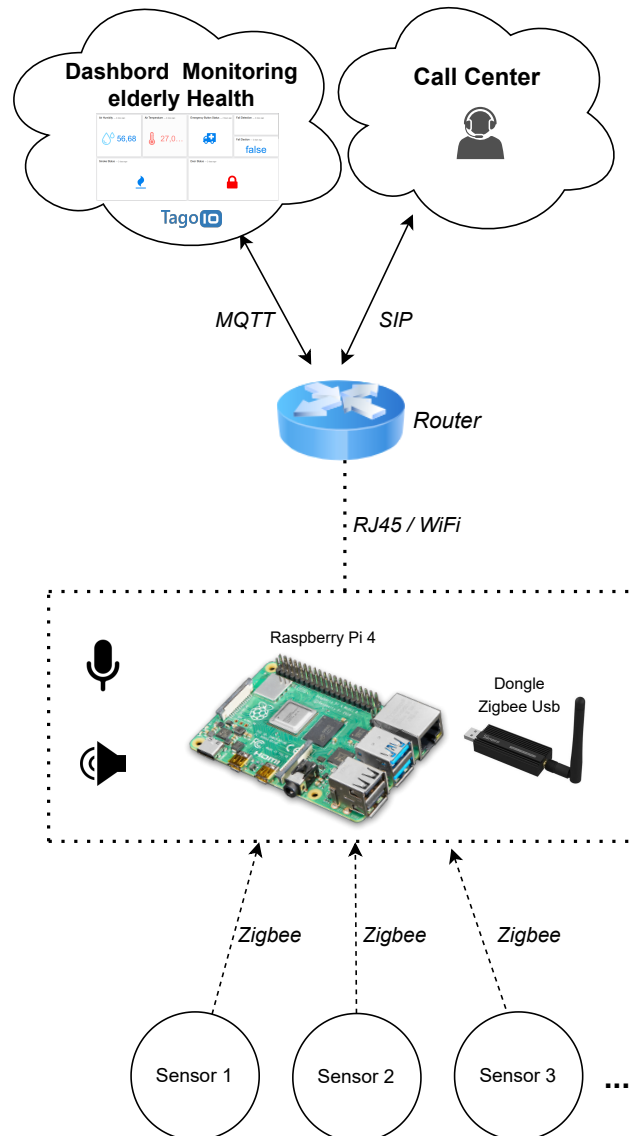
Para visualização dos dados em tempo real, desenvolveremos um parser dentro da plataforma *TagoIO* e montaremos uma tela amigável para acompanhar os eventos dos sensores instalados no ambiente de operação.

A nível de servidores, será utilizado um serviço de nuvem como *AWS* ou *Azure* para instalação de um *Asterisk* como central *Private Automatic Branch Exchange over Internet Protocol (PABX IP)*. E para a tratativa dos dados enviados pelo *gateway* através do protocolo *MQTT*, utilizaremos o próprio *broker MQTT* da *Tago.IO*.

Para salvar as configurações da interface web do produto e outros dados importantes, usaremos um simples banco de dados *MYSQL* previamente instalado no *Linux* da *Raspberry Pi 4*.

A topologia do sistema a ser projetado pode ser visto na [Figura 5](#).

Figura 5 – Topologia do sistema proposto



Fonte: AUTOR

### 3.1.1 Descrição dos componentes

Nesta seção será detalhada os componentes utilizados para construir o protótipo.

#### 3.1.1.1 Raspberry Pi 4

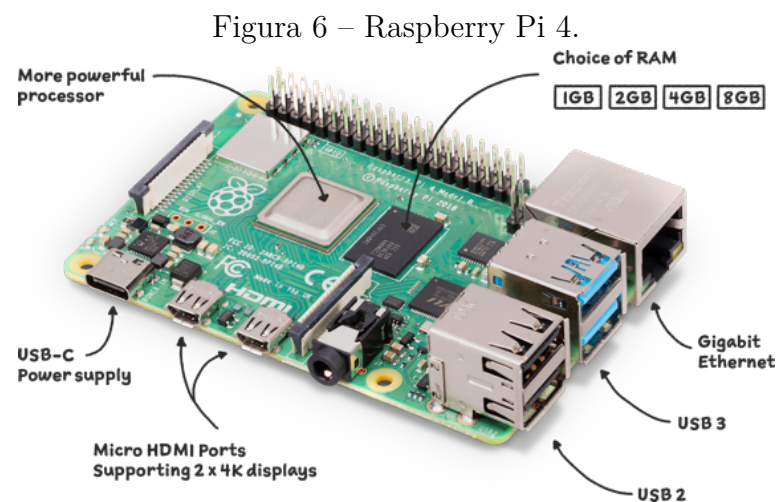
Raspberry Pi é uma linha de produtos de micro computadores da *Raspberry Pi Foundation*, uma entidade de caridade do Reino Unido. Que tem o intuito de oferecer um hardware de baixo custo podendo ser comparados em desempenho com computadores. (FOUNDATION, 2012).

Para nosso desenvolvimento do protótipo selecionamos o dispositivo *Raspberry Pi 4* com 4GB de memória RAM e tem como outras características técnicas:



- Rede: Porta de rede Gigabit Ethernet, WiFi, Bluetooth.
- USB: 4 portas.
- Processador: Quad core 64-bit ARM-Cortex A72.
- Clock: 1.5GHz.
- Possui um slot para SD Card.
- Alimentação: 5V.
- Vídeo: 2 portas Micro HDMI 4k.
- GPIO: 40 pinos.

A placa pode e seus periféricos podem ser visualizados na [Figura 6](#):



Fonte: (RASPBERRY..., )

O dispositivo oferece 40 pinos, tendo pinos para alimentação, para aterramento, para comunicação serial e outros que podem ser configurados como entrada ou saída de dados. Os pinos **General Purpose Input/Output (GPIO)** entendem como alto(1) quando um nível elétrico é entre 1,8V e 3V. E baixo(0) quando é menor que 1,8V. O *pinout* e suas configurações podem ser observados na [Figura 7](#):

Figura 7 – Pinout da Raspberry Pi 4.

PIN	NAME		NAME	PIN
01	3.3V DC Power	Red	5V DC Power	02
03	GPIO02 (SDA1, I <sup>2</sup> C)	Blue	5V DC Power	04
05	GPIO03 (SDL1, I <sup>2</sup> C)	Blue	Ground	06
07	GPIO04 (GPCLK0)	Green	GPIO14 (TXD0, UART)	08
09	Ground	Black	GPIO15 (RXD0, UART)	10
11	GPIO17	Green	GPIO18(PWM0)	12
13	GPIO27	Green	Ground	14
15	GPIO22	Green	GPIO23	16
17	3.3V DC Power	Red	GPIO24	18
19	GPIO10 (SP10_MOSI)	Purple	Ground	20
21	GPIO09 (SP10_MISO)	Purple	GPIO25	22
23	GPIO11 (SP10_CLK)	Purple	GPIO08 (SPI0_CE0_N)	24
25	Ground	Black	GPIO07 (SPI0_CE1_N)	26
27	GPIO00 (SDA0, I <sup>2</sup> C)	Yellow	GPIO07 (SCL0, I <sup>2</sup> C)	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12 (PWM0)	32
33	GPIO13 (PWM1)	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40

Fonte: (KHAN, 2022)

### 3.1.1.2 SONOFF ZigBee 3.0

SONOFF Zigbee 3.0 USB Dongle(Figura 8) é um adaptador USB Zigbee universal. Segundo a (SONOFF, 2022), ele pode ser usado como um *gateway Zigbee* universal em aplicações de *Smart Home* ou outras plataformas de código aberto via *Zigbee2MQTT* para controlar localmente todos os seus subdispositivos *Zigbee*, para que você não precise investir nos *hubs Zigbee* para diferentes marcas. Então com esse adaptador USB mais a *Raspberry Pi* conseguimos construir um *gateway Zigbee* para processar sinais de quaisquer sensores *Zigbee* de mercado.

Figura 8 – SONOFF ZigBee 3.0.



Fonte: (SONOFF, 2022)

O produto possui as seguintes características:

- CC2652P + CP2102N.
- +20dBm de ganho de saída.
- Antena externa com conector *SMA*.
- Case de alumínio.
- Frequência de operação 2,405 - 2,485 GHz.

#### 3.1.1.3 Alto falante e microfone

Para a comunicação de áudio usaremos um microfone para capturar o áudio no produto e um alto falante com interface de áudio *P2* e alimentado através da porta USB da *Raspberry Pi 4*. Desta forma poderemos fazer a comunicação de voz com o idoso.

#### 3.1.1.4 Sensores RF *ZigBee*

Para a solução proposta vamos utilizar alguns sensores *ZigBee* de mercado para nossa solução. Desde sensor de vibração, para processar quedas do idoso, botão/colar de emergência para o idoso iniciar um evento de emergência e outros sensores amplamente utilizado em soluções de *Smart Home* como sensor de porta, sensor de fumaça, sensor de gás de cozinha, etc.

#### 3.1.1.5 Cliente SIP

Para geração das chamadas de emergência *SIP*, utilizaremos o *softphone Linphone*. O mesmo oferece um *daemon* capaz de ser executado a partir de comandos do *Bash Linux*.

## 3.2 Desenvolvimento do sistema

Nessa seção será detalhado o desenvolvimento do sistema como um todo.

### 3.2.1 Desenvolvimento do gateway

#### 3.2.1.1 Instalação do sistema operacional

Para primeira parte do desenvolvimento, utilizamos o hardware *Raspberry Pi 4* para a instalação do sistema operacional *Linux*.

#### 3.2.1.2 Instalação do softphone Linphone

Via terminal instalamos o *Linphone* trata-se de um *softphone opensource*. A função principal deste software será a intercomunicação **SIP** com um **PABX IP**. Com o uso de um microfone e um alto falante na interface de áudio da placa *Raspberry Pi 4* realizamos e receber chamadas.

Usaremos o *daemon* para console *Linphonec*, para controlar as ações de telefonia **SIP**. As configurações do *softphone* é realizada a partir do arquivo *".linphonec"*. Como por exemplo as configurações de ramal **SIP**, de volume, seleção das interfaces de áudio para entrada e saída de áudio.

#### 3.2.1.3 Monitoração da GPIO - botoeira

Para geração de chamadas, o produto contém um botão para geração da chamada para um número pré-configurado na interface web. Podemos utilizar a biblioteca *wiringpi* do *Python* para ficar monitorando o estado dos botões que o produto possuirá, onde quando pressionado será originada uma chamada *SIP* para uma central de monitoramento.

#### 3.2.1.4 Instalação Zigbee2MQTT

Para o tratamento dos dados que chegam no receptor *Zigbee USB* e o envio para a nuvem através do protocolo **MQTT**, utilizamos o aplicativo *Zigbee2MQTT* que conecta redes *Zigbee* a redes **MQTT**. E já tem total integração com o *Dongle SONOFF ZigBee 3.0*.

### 3.2.2 Desenvolvimento da interface

#### 3.2.2.1 Interface de configuração do produto

A interface web de configuração do produto será desenvolvida utilizando a linguagem *Python* para o *backend* e o *frontend* em Linguagem de Marcação de HiperTexto (**HTML**). As telas para configuração do produto contemplará a possibilidade de alterações nas

configurações SIP, configurações de rede, alterar senha de administrador web, também uma tela para abrir e fechar a rede de pareamento *Zigbee* e reiniciar o dispositivo.

Com todas configurações aplicadas, o *software* de leitura dos botões e da interface web do produto será iniciada no *boot* do sistema operacional.

### 3.2.2.2 Interface de monitoração - Dashbord

Para a visualização dos dados coletados pelos sensores em tempo real, será exibido em um *dashboard* futuramente desenvolvido na *TagoIO*. A mesma se comportará como um *Broker MQTT*.

### 3.2.3 desenvolvimento do protótipo

Após integrar todas as aplicações e testa-las em conjunto. Será adquirido um case para comportar todos os componentes do sistema a fim de ter a aparência de um produto final. O produto será instalado na residência dos pais do autor junto com sensores para aplicar um teste de campo no protótipo desenvolvido.

## 3.3 Cronograma de execução

Figura 9 – Cronograma de desenvolvimento

CRONOGRAMA DE DESENVOLVIMENTO DE PRODUTO		Data de Início do Projeto																																			
		7/6/2022																																			
Produto: Sistema de monitoramento e alerta emergenciais para idosos																																					
Atividade	Responsável	Sep-22				Oct-22				Nov-22				Dec-22				Jan-23				Feb-23				Mar-23				Apr-23				May-23			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Estudo de Viabilidade	Andrey	■	■	■	■																																
Estudo das tecnologias	Andrey					■	■	■	■	■	■	■	■																								
Estudo dos hardwares	Andrey													■	■	■	■																				
Adquirir componentes	Andrey																	■	■	■	■																
Hardware - Mecânico	Andrey																					■	■	■	■												
Produção de protótipos	Andrey																									■	■	■	■								
Dev. Software	Andrey																													■	■	■	■				
Testes homolog produto	Andrey																																				
Documentação Técnica	Andrey																																				

Fonte: AUTOR

## 4 CONCLUSÕES

Com o protótipo em suas perfeitas condições de funcionamento, temos um produto capaz de monitorar uma porção de diferentes sensores *Zigbee* de mercado, proporcionando um produto para monitoramento de idosos e alertas emergências gerados no ambiente de instalação, além disso funcionando como um produto *IP* para tele assistência ao idoso.

A solução desenvolvida oferece um *dashboard* para monitoração em tempo real na nuvem e uma interface de configuração amigável para configurações de rede, *SIP* e adição de sensores. Desta forma será possível fazer um atendimento mais ágil ao idoso, evitar possíveis acidentes.

Os objetivos da proposta inicial foram atingidos com sucesso, após ter algumas mudanças que surgiram no caminho na implementação do projeto, como a mudança do hardware de desenvolvimento, alterando o uso de uma *Orangi Pi Zero LTS* para uma *Raspberry Pi 4* devido alguns problemas de desempenho provido das aplicações operando em paralelo.

Para termos um produto a nível comercial ainda é necessário um grande esforço de desenvolvimento e para um novo importante recurso seria a implementação de uma alternativa de *fallback* para o envio dos dados coletados e comunicação. Por exemplo, fazendo o uso de um modem 4G.

# REFERÊNCIAS

ANDREIS, I. Estudo comparativo entre bibliotecas de implementação para o protocolo sip. 2010. Disponível em: <<https://www.lume.ufrgs.br/handle/10183/28312>>. Acesso em: 1 dez 2022. 11

ANDRIGHETTO, E. *Sistema de processamento de sinais biomédicos: rede wireless ZigBee com aplicação do padrão IEEE 802.15.4*. 2008. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/92085>>. Acesso em: 23 nov 2022. 6

CONCEIÇÃO, W. N. E. da; COSTA, R. M. de R. Análise do protocolo mqtt para comunicação iot através de um cenário de comunicação. *Caderno de Estudos em Sistemas de Informação*, v. 5, n. 2, 2019. Acesso em: 29 nov 2022. 8

FOUNDATION, R. P. *Raspberry Pi Foundation – About us*. 2012. Disponível em: <<https://www.raspberrypi.org/about/>>. Acesso em: 15 nov 2022. 15

IBM. *Padrões e requisitos do MQTT - Documentação da IBM*. 2021. Disponível em: <<https://www.ibm.com/docs/pt-br/maximo-monitor/8.2.0?topic=messaging-standards-requirements>>. Acesso em: 29 nov 2022. 10

KHAN, A. *Raspberry Pi 4 GPIO Pinout*. 2022. Disponível em: <<https://linuxhint.com/gpio-pinout-raspberry-pi/>>. Acesso em: 21 nov 2022. 17

KINDERMANN, L. de M.; SANTOS, E. L. F. dos. *ESTUDO SOBRE O PADRÃO ZIGBEE E DESENVOLVIMENTO DE SISTEMA APLICADO EM AUTOMAÇÃO RESIDENCIAL*. Revista Ilha Digital, 2012. Disponível em: <<https://ilhadigital.florianopolis.ifsc.edu.br/index.php/ilhadigital/article/view/35/34>>. Acesso em: 23 nov 2022. 6, 7, 8

MARTINEZ, J. G. A.; JR, J. M. da S. Desenvolvimento de uma aplicação voip baseada no protocolo sip. Universidade do Estado do Amazonas, 2011. Disponível em: <<http://tiagodemelo.info/monografias/2011/tcc-joao-martinez.pdf>>. Acesso em: 1 dez 2022. 11, 12

MARTINS, I. R.; ZEM, J. L. Estudo dos protocolos de comunicação mqtt e coap para aplicações machine-to-machine e internet das coisas. Faculdade de Tecnologia de Americana, 2015. Acesso em: 29 nov 2022. 8, 9, 10

OMS. *RELATÓRIO MUNDIAL DE ENVELHECIMENTO E SAUDE*. 2015. 1-30 p. Disponível em: <<https://sbgg.org.br/wp-content/uploads/2015/10/OMS-ENVELHECIMENTO-2015-port.pdf>>. Acesso em: 18 nov 2022. 4

RASPBERRY Pi 4. Disponível em: <<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>>. Acesso em: 15 nov 2022. 16

ROSENBERG, J. et al. *SIP: session initiation protocol*. 2002. Disponível em: <<https://dl.acm.org/doi/pdf/10.17487/RFC3261>>. Acesso em: 1 dez 2022. 11

- RUST, T. Fuzzing zigbee using z-stack. 2022. Disponível em: <[http://www.cs.ru.nl/bachelors-theses/2022/Tom\\_Rust\\_\\_\\_\\_1040068\\_\\_\\_\\_Fuzzing\\_Zigbee\\_using\\_Z-Stack.pdf](http://www.cs.ru.nl/bachelors-theses/2022/Tom_Rust____1040068____Fuzzing_Zigbee_using_Z-Stack.pdf)>. Acesso em: 1 dez 2022. 11
- SANTOS, A. V. *UMA SOLUÇÃO SDN PARA A COMUNICAÇÃO MESH DE NÓS EM UMA REDE ZIGBEE PADRÃO 802.15.4*. Dissertação (Mestrado) — UNIVERSIDADE FEDERAL DO CEARÁ, Ceará, 2015. Disponível em: <<https://repositorio.ufc.br/handle/riufc/25000>>. Acesso em: 23 nov 2022. 7
- SONOFF. *SONOFF Zigbee 3.0 USB Dongle Plus-P-SONOFF Oficial*. 2022. Disponível em: <<https://sonoff.tech/product/diy-smart-switch/sonoff-zigbee-3-0-usb-dongle-plus-p/>>. Acesso em: 18 nov 2022. 17, 18
- SOUSA, J. P.; CARRAPATOSO, E. Uma arquitetura iptel baseada no protocolo sip. In: *6ª CONFERÊNCIA SOBRE REDES DE COMPUTADORES (CRC'2003–Bragança, Portugal)*. Universidade de Coimbra, Instituto Politécnico de Bragança, 2003. Disponível em: <<https://bibliotecadigital.ipb.pt/handle/10198/7162>>. Acesso em: 1 dez 2022. 12
- STOLL, G. R. *O que é este tal do zigbee*. UTC Journal-Smart Utilities Networks, Special Issue, 2008. Disponível em: <[http://www.mecomp.com.br/rumo/o\\_que\\_e\\_este\\_tal\\_do\\_zigbee\\_3\\_.pdf](http://www.mecomp.com.br/rumo/o_que_e_este_tal_do_zigbee_3_.pdf)>. Acesso em: 22 nov 2022. 6
- TAGIOIO. 2022. Disponível em: <<https://tagio.io/>>. Acesso em: 1 dez 2022. 13
- VALENZUELA, W. A. V. et al. *Uma abordagem teórica e prática em um protocolo para IoT*. 2021. 43405–43418 p. Acesso em: 29 nov 2022. 9
- XAVIER, R. C. *Conheça a TagoIO, a primeira ferramenta cloud para desenvolvimento de solução IoT homologada Khomp*. 2019. Disponível em: <<https://www.khomp.com/pt/tagioio-solucao-iot/>>. Acesso em: 1 dez 2022. 13