



DOMÓTICA COM CONCEITOS DE IOT UTILIZANDO AGENTE DIALOGFLOW (DIANA) PARA RECONHECIMENTO DE FALA

Lemuel Henner Verçosa Santos¹; Alexandre Edson Silva Pereira¹; Elias José de Melo Silva¹; João Guilherme Silva Macedo¹; René Marcelino Abritta Teixeira¹.

RESUMO: Esta pesquisa tem o objetivo de alcançar menores tempos de resposta e maiores taxas de acerto em uma automação residencial baseada em comandos de voz e texto com variações de sintaxes e utilização de palavras sinônimas, feitos através de múltiplas plataformas. A combinação de ferramentas testadas (Dialogflow + Python + Arduino / ESP) demonstrou resultados satisfatórios, com o valor médio de 2,308 segundos para o tempo de resposta e de 94,16% para a taxa de acerto.

PALAVRAS-CHAVE: Automação residencial; Reconhecimento de voz; Dialogflow; Python; Arduino / ESP

DOMOTICS WITH IOT CONCEPTS USING DIALOGFLOW AGENT (DIANA) FOR SPEECH RECOGNITION

ABSTRACT: This research has the goal to achieve lower response times and better accuracy rate in a home automation based on voice and text commands with syntax variation and synonyms words, via multiple platforms. The tools' combination tested (Dialogflow + Python + Arduino / ESP) showed satisfactory results, with the average value of 2,308 seconds for the response time and 94,16% for the accuracy rate.

KEYWORDS – Home automation, voice recognition, Dialogflow, Python, Arduino / ESP.

¹ Centro Universitário das Faculdade Metropolitanas Unidas (FMU)

INTRODUÇÃO

O campo da automação residencial tem se desenvolvido nos últimos anos a partir de novas tecnologias em busca de comodidade e acessibilidade dos moradores. No entanto, algumas dessas tecnologias como o reconhecimento de voz, muitas vezes não são inclusas em sistemas de automação, pois apresentam dificuldades de aplicação e utilização em relação a ruídos, entendimento de ritmos, tons e idioma da fala.

Apesar disso, o reconhecimento de fala pode servir como uma ferramenta proveitosa nas rotinas de seus usuários, pois, associada a outros recursos, torna possível o controle de uma casa pelo morador através de um comando de voz e sem a necessidade de ação humana palpável. Por conta disso, também é muito útil para pessoas com deficiência ou mobilidade reduzida. Para que seja aplicado reconhecimento de fala em um projeto, normalmente é utilizada Inteligência Artificial (IA) para realizar o processamento do áudio. Por conta de sua complexidade e necessidade de grande poder de processamento, a IA é desenvolvida por empresas de tecnologia (Microsoft, Google, IBM etc.) e disponibilizada na forma de webservice através de Application Programming Interfaces (APIs), para ser consumida por desenvolvedores de sistemas em seus programas.

Neste artigo será apresentado o funcionamento e resultados de um projeto de automação residencial com reconhecimento de voz. Foi utilizada a API DialogFlow integrada à uma programação em Python para possibilitar a comunicação dos usuários com a casa via microfone local, Google Assistente ou Whatsapp Web, aceitando comandos com diferentes sintaxes (estrutura e ordem das palavras na frase) e palavras sinônimas (mesmo significado) e utilizado um microcontrolador para o acionamento dos dispositivos eletrônicos da casa. Também foi implementado um supervisor, onde é possível verificar o comportamento de dispositivos conectados à Internet, que demonstram o conceito conhecido como Internet das Coisas (IoT - Internet of Things). Foram mensurados no projeto os parâmetros de tempo de resposta dos comandos (tempo até a execução do comando) e a taxa de acerto na compreensão dos comandos.

Será mostrado um estudo simplificado de três monografias de trabalhos de conclusão de curso voltados a reconhecimento de voz e um artigo de comparação de diferentes API de reconhecimento de fala.

A monografia de Freitas (2019) propõe um sistema baseado em um aplicativo desenvolvido na plataforma MIT AppInventor, que faz uso direto da API Google Cloud Speech-to-Text para o reconhecimento de voz e aciona dispositivos através de um Arduino conectado ao smartphone por Bluetooth. Este sistema foi apresentado em forma de um protótipo utilizando LEDs para simular as cargas acionadas, diferentemente do trabalho proposto nesse artigo, em que foi realizada instalação de equipamentos em uma casa real, além de utilizar conexão via Wi-Fi para comunicação dos dispositivos. Os resultados obtidos por Freitas se referem às taxas de acerto dos comandos em quatro cenários, considerando a distância entre o usuário e microfone (o mais próximo da boca possível, considerado como 0 metros, ou 1 metro) e o ruído do ambiente (com ou sem ruído) , além da taxa total de acertos. Os valores obtidos podem ser vistos na tabela 01.

Tabela 1 – Resultados do trabalho de Freitas

Taxa de acerto (%)			
Ambiente Distância (m)	Com ruído	Sem ruído	Total
0	97,22	100	98,61
1	80,56	99,44	90
Total	88,89	99,72	94,3

Já o trabalho de conclusão de curso de Neto (2018) sugere um protótipo de automação residencial por comando de voz dado a um dispositivo móvel com utilização da assistente pessoal Amazon Alexa, para interpretação dos comandos, e o Arduino Ameba, para controle dos dispositivos da casa (lâmpadas e porta) e aferição de temperatura através de sensores. Os testes foram direcionados para o tempo de resposta (tempo entre o comando de voz e o acionamento), obtendo como resultados 4 segundos de tempo, mas ressaltando a sua variabilidade de acordo com a velocidade da conexão. O trabalho de Neto possui algumas limitações: o sistema foi aplicado com comandos em inglês (em

razão da limitação de idioma que a Alexa tinha na época) e não é capaz de entender variações de sintaxe nos comandos por voz. Além disso, o projeto não foi aplicado em um ambiente autêntico, mas desenvolvido em uma maquete de casa.

O trabalho de conclusão de curso de Goll (2017) apresenta uma aplicação de inteligência virtual (apelidada de VIKI) para realizar o reconhecimento de um comando de voz do usuário e transformar esses comandos recebidos em uma ação para acionar uma luz e controlar uma televisão. Para isso, desenvolveu-se uma aplicação em C# para fazer a integração dos sistemas, API Microsoft Speech Platform para realizar o reconhecimento e síntese de voz e o ESP8266 como microcontrolador. VIKI utiliza palavras-chaves fixadas e pré-programadas para o reconhecimento dos comandos de voz, ao contrário da proposta do projeto aqui apresentado, que utiliza frases de treinamento para uma inteligência artificial de machine learning interpretar os comandos e variações. Quanto aos testes, Goll optou por realizar testes qualitativos, voltados para a experiência do usuário e usabilidade do sistema em um ambiente real e em um ambiente simulado, não incluindo testes com métricas precisas, como tempo de resposta, acurácia do reconhecimento ou taxa de acertos. Em seus resultados, ele compara as principais características do trabalho (reconhecimento de voz por comandos e sentenças específicas, reconhecimento e síntese do idioma português brasileiro, síntese de voz, capacidade de diálogo básico e integração com Internet das Coisas) com trabalhos correlatos.

Por fim, a pesquisa de Bohouta e Kěpuska (2017), publicada em um artigo do Instituto de Tecnologia da Flórida, propõe identificar dentre três sistemas de reconhecimento de fala (Microsoft API e Google API e CMU Sphinx) seria mais eficaz. Utilizando como métrica a taxa de erro de palavras (Word Error Rate – WER, em inglês) e acurácia nas frases de áudio e através de testes com programa em Java e amostras de áudios gravados em inglês, os autores determinaram que a Google API possui melhor modelo acústico e de idioma, com WER de 9% e acurácia de 91% (conforme pode ser visto na tabela 02).

Tabela 2 – Resultados obtidos por Bohouta e Këpuska

WER e Acurácia (%)		
Plataforma	Acurácia	WER
Google API	91	9
Microsoft API	82	18
CMU Sphynx	63	37

De acordo com os resultados apresentados por Bohouta e Kepüska, nesta pesquisa será baseada na API DialogFlow da GoogleCloud.

A Dialogflow é uma plataforma de processamento de linguagem natural que facilita o design e a integração de uma interface conversacional do usuário com apps para dispositivos móveis, aplicativos da Web, bots, sistemas de unidade de resposta audível etc. Oferece diversas tecnologias integradas, como por exemplo o aprendizado de máquina, que permite que a API reconheça a intenção do usuário e receba parâmetros como data e hora. Também é possível usar a sua tecnologia de reconhecer diversas línguas, por possuir o Google Tradutor, e realizar integrações com o Google Assistente assim entendendo frases em português, garantindo maior acessibilidade e usabilidade do sistema.

METODOLOGIA

Para realização dos testes, foram utilizados o microcontrolador Arduino, o SOC ESP01, além de módulos relé, dimmer e sensor de tensão alternada. Na parte de software, foi utilizada a linguagem Python, o supervisor ScadaBR, banco de dados SQLite, a API Dialogflow e IDEs para codificação. Os dispositivos do sistema foram conectados entre si utilizando a Internet e cabos seriais, fazendo uso dos protocolos de comunicação TCP/IP, Modbus/TCP e UART.

RESULTADOS E DISCUSSÃO

De acordo com os resultados apresentados por Bohouta e Kepüska, nesta pesquisa será utilizada a API DialogFlow da GoogleCloud. Com base nos trabalhos de Goll (2017), Freitas (2019) e Neto (2018) a metodologia do funcionamento do sistema em estudo ocorre em quatro etapas. A primeira etapa

é a de reconhecimento de voz, quando o morador utiliza a sua voz para realizar comandos e assim acionar algum dispositivo eletrônico de acordo com o comando solicitado. A segunda etapa está relacionada ao processamento da voz, em que a API Dialogflow irá receber a informação dada pelo usuário em áudio e converter tal informação para uma linguagem de máquina. Já na terceira etapa, a Dialogflow envia para o programa em Python as informações necessárias para realizar os comandos de acordo com as informações processadas. Na quarta etapa, o Arduino recebe essas informações e aciona o dispositivo eletrônico de acordo com a solicitação feita pelo usuário no início do processo.

A fim de verificar a factibilidade do sistema proposto foram feitos alguns testes mínimos de seu funcionamento. A plataforma do Dialogflow exige a criação de um agente virtual, que foi nomeado como Diana. No console do Dialogflow, na criação da Diana, são definidas “intents”, as intenções ou assuntos que o usuário pode transmitir, e as frases associadas a essas “intents”, utilizadas para o treinamento da inteligência artificial, permitindo reconhecer e entender variações de frases (inclusive com erros de pronúncia).

O sistema de testes foi desenvolvido utilizando a linguagem Python. Este sistema comunica-se com o Dialogflow e com o módulo Arduino. Os testes foram conduzidos utilizando uma lâmpada e um ventilador como objeto de automação. Também foram realizadas medições de temperatura e umidade. Na figura 1 é mostrado o esquema elétrico dos acionamentos do ventilador e da lâmpada.

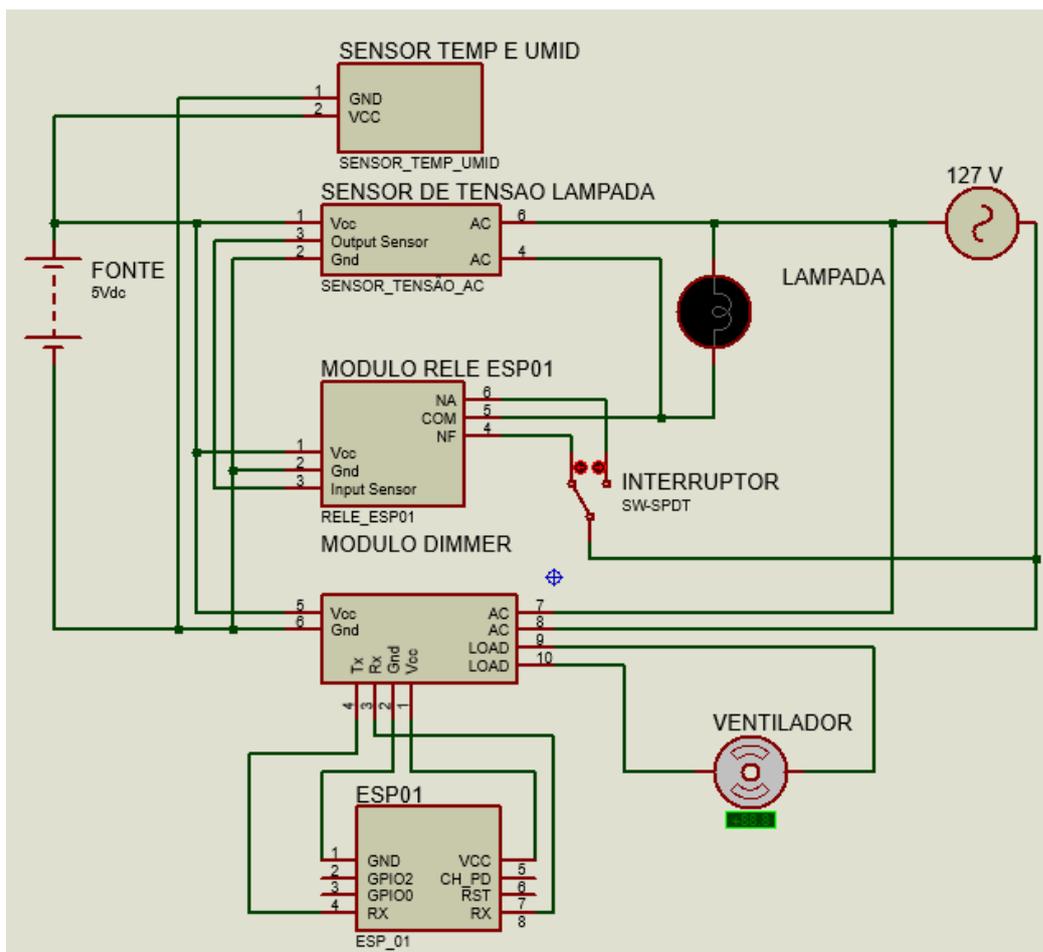


FIGURA 1 – ESQUEMA ELÉTRICO DE ACIONAMENTO
 FONTE: Desenvolvido pelos autores

O script Python é dividido em duas partes: a de entrada do áudio do microfone local e a interface Arduino / Dialogflow. O script do microfone local ativa o microfone e aguarda o usuário falar. Para evitar chamadas indesejáveis da API, há uma condicional para iniciar a interação com a Diana, verificando frases chaves como “Ok Casa” e “Ok Diana”. Após isso, o programa ouve o comando do usuário, sintetiza o áudio e faz a chamada / solicita a API Dialogflow, que se comunica com o programa de interface Arduino / Dialogflow para executar o comando solicitado. O script então fica aguardando o retorno da API e, ao receber a resposta, ativa o alto-falante, reproduz o áudio de resposta e volta a ouvir o usuário, esperando a fala das frases chaves. A figura 2 mostra o fluxograma do programa do microfone local.

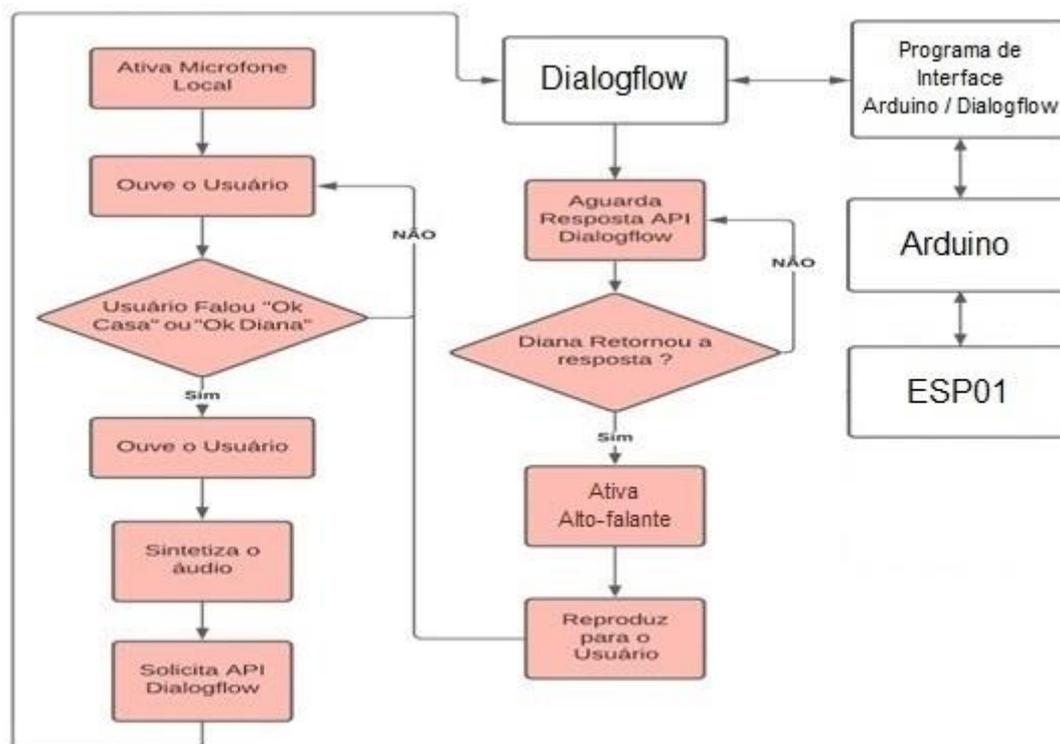


FIGURA 2 – FLUXOGRAMA DO PROGRAMA DE INTERFACE MICROFONE LOCAL / DIALOGFLOW.

FONTE: Desenvolvido pelos autores

O programa de interface Arduino / Dialogflow aguarda a chamada de webhook até que a Diana (agente Dialogflow) solicite acesso. Quando isso ocorre, o programa prepara as informações recebidas e, caso seja um comando, envia a informação para o Arduino. O programa então aguarda o retorno do Arduino e quando o recebe, trata as informações recebidas, armazena informações da solicitação (plataforma utilizada, tempo de resposta e tipo de mensagem, texto ou voz) em um banco de dado, retorna a resposta do Arduino para o Dialogflow e aguarda nova chamada de webhook. A figura 3 mostra o fluxograma deste processo.

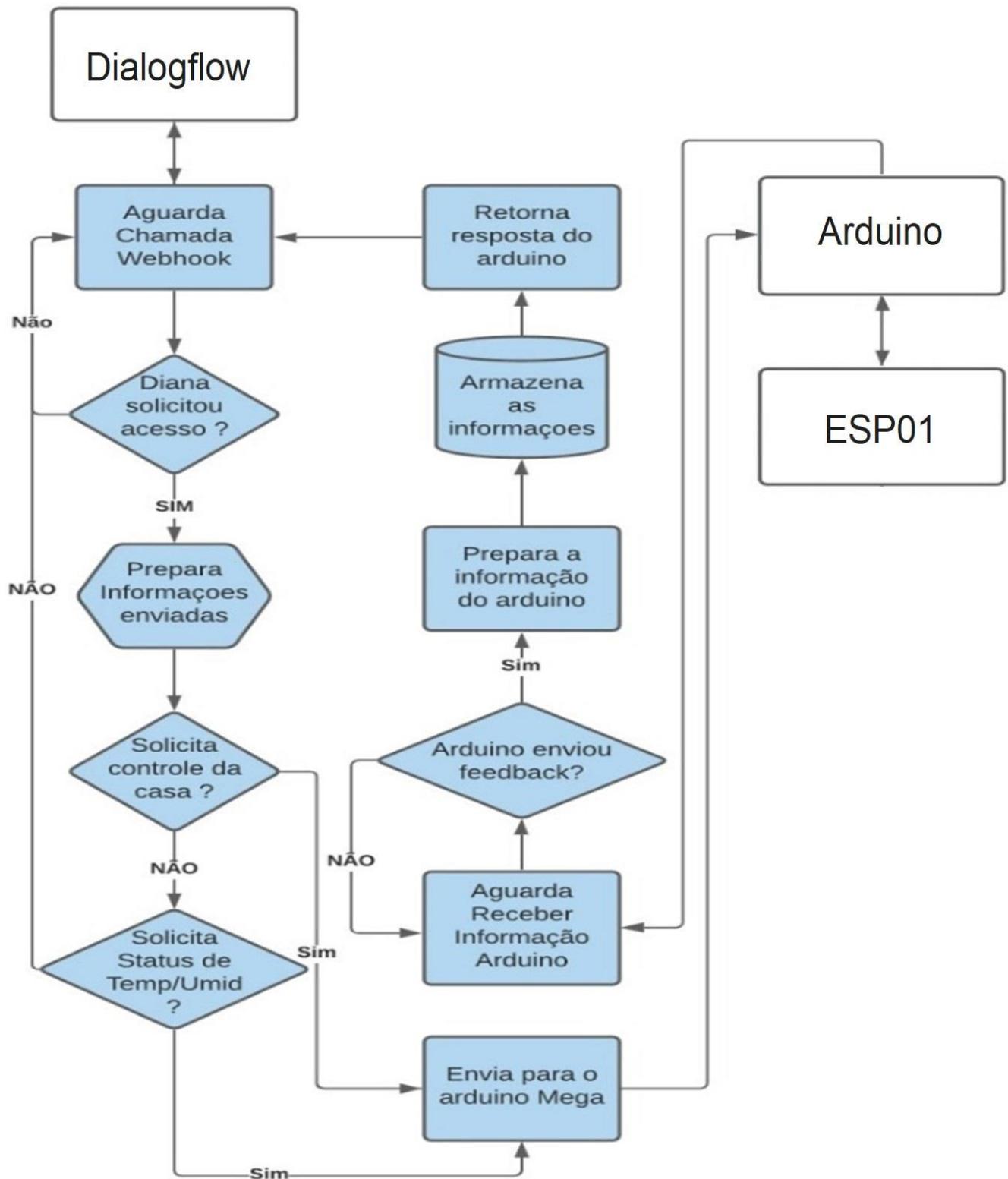


FIGURA 3 - FLUXOGRAMA DO PROGRAMA DE INTERFACE ARDUINO / DIALOGFLOW.

FONTE: Desenvolvido pelos autores

A programação do Arduino inicializa o display LCD, shield Ethernet e a conexão via Protocolo Modbus/TCP e então aguarda a solicitação da Diana. Ao receber a solicitação, é realizada uma verificação do tipo de comando recebido (controle da luz da sala, da luz do quarto, do ventilador do quarto e leitura de temperatura / umidade, nessa ordem) e enviado o comando para o ESP01 correspondente. O programa do Arduino passa a aguardar a resposta do ESP01 acionado, retorna a informação para o Dialogflow e aguarda uma nova solicitação deste. O fluxograma da figura 4 exhibe os passos do processo.

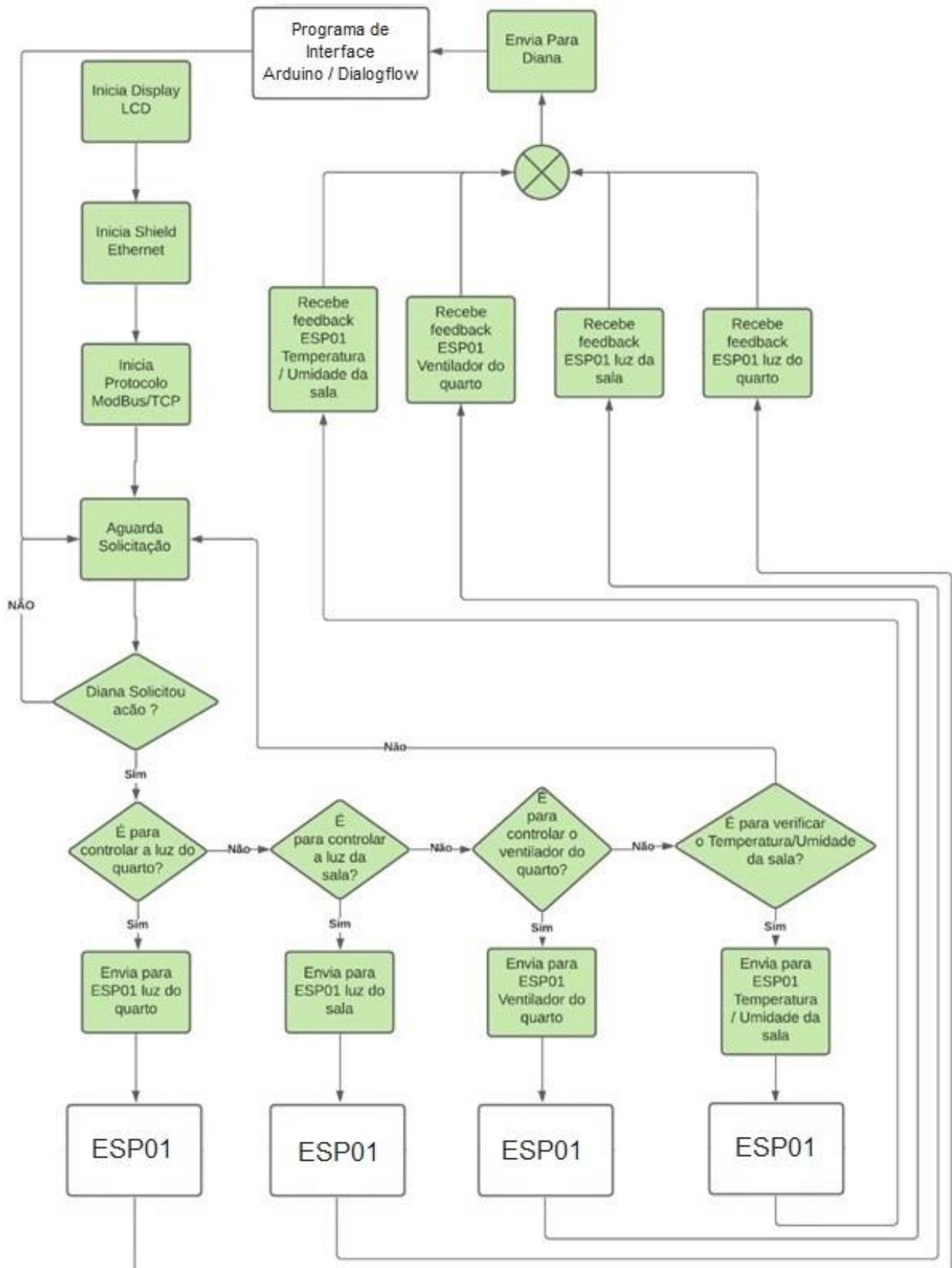


FIGURA 4 – FLUXOGRAMA DA PROGRAMAÇÃO DO ARDUINO.

FONTE: Desenvolvido pelos autores

A programação dos ESP01 funciona de maneira análoga à parte de verificação da programação do Arduino. Caso o comando recebido do Arduino seja de controle (lâmpada ou ventilador), é verificado o tipo de ação (liga ou desliga), verificado o status do dispositivo, executada a ação de acordo e enviada uma resposta para o Arduino. Caso o comando recebido do Arduino seja de verificação (status dos dispositivos e medição de temperatura / umidade), é realizada a verificação e enviada a resposta para o Arduino. Na figura 5 é mostrado o fluxograma do processo.

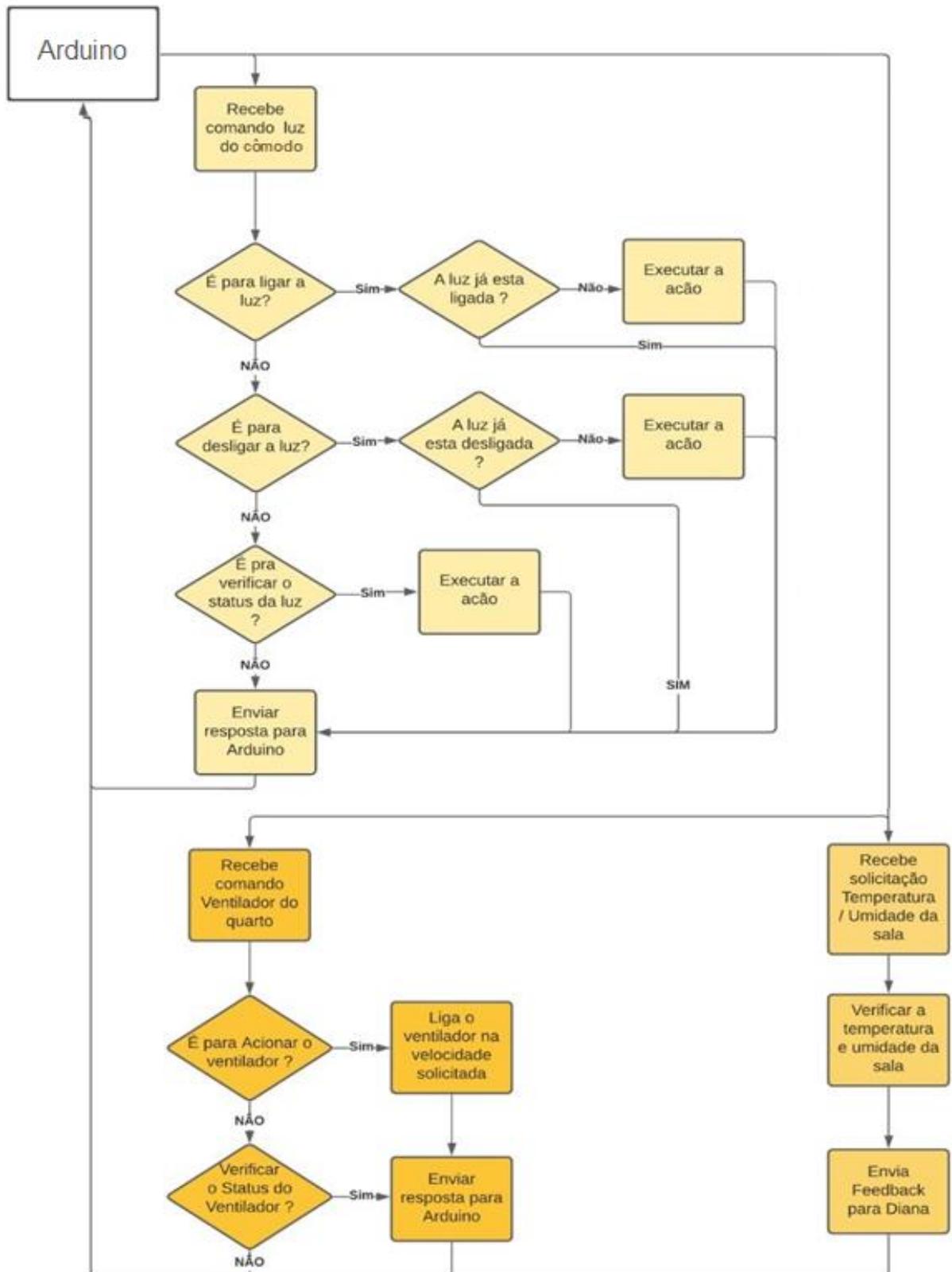


FIGURA 5 – FLUXOGRAMA DA PROGRAMAÇÃO DOS ESP01.

FONTE: Desenvolvido pelos autores

De acordo com o objetivo de disponibilizar ao usuário diversas plataformas de comunicação, existe a integração com o WhatsApp Web, utilizando programação em Python, que possibilita manipular a página Web de qualquer navegador como se fosse um ser humano e assim enviar / receber mensagens de áudio / texto. Para isso, também foi criado um contato (utilizando chip SIM de uma operadora) para a Diana, possibilitando a comunicação com o usuário. A figura 6 mostra o fluxograma do processo do Whatsapp.

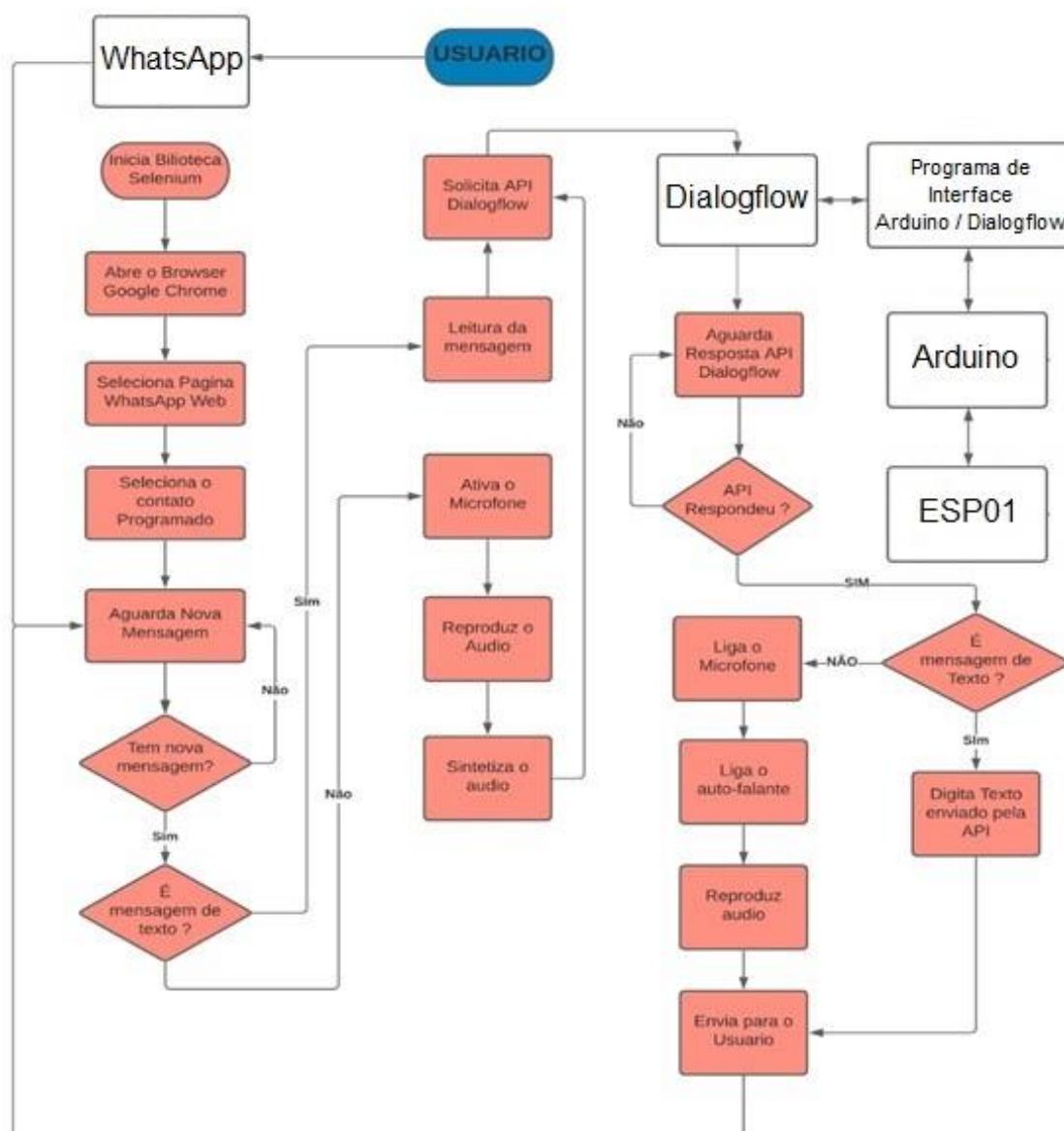


FIGURA 6 – FLUXOGRAMA DO PROGRAMA DE INTERFACE WHATSAPP / DIALOGFLOW.

FONTE: Desenvolvido pelos autores

Outra forma de comunicação aplicada foi a integração direta entre o DialogFlow e o Google Assistente (figura 7), com ambos utilizando a mesma tecnologia de reconhecimento de voz. Dessa maneira, é possível a implementação de um método de interação mais direto com a API.

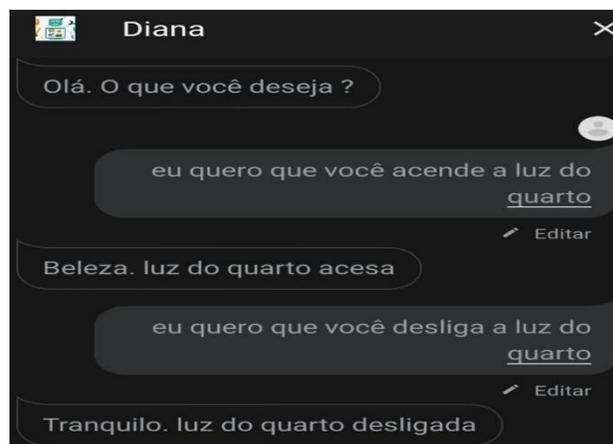
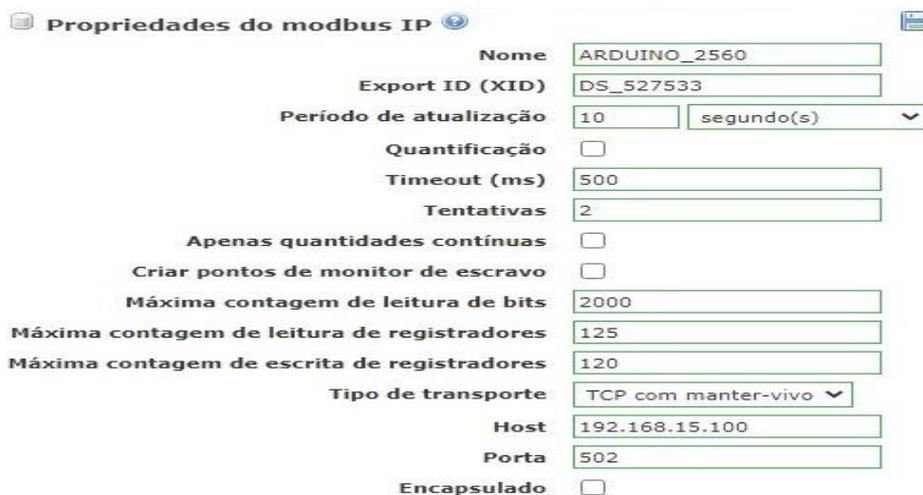


FIGURA 7 – INTERFACE DA DIANA VIA GOOGLE ASSISTENTE.

FONTE: Desenvolvido pelos autores

No intuito de garantir uma forma de supervisão geral da residência para o usuário, foi utilizado o ScadaBR. Utilizou-se o Arduino 2560 para receber todas as informações via rede TCP/IP dos microcontroladores ESP-8266 distribuídos pela residência e enviá-las para o supervisor via protocolo Modbus-TCP/IP. No ScadaBR foi configurada a rede Modbus para fazer a leitura dos pontos fornecidos pelo Arduino.

Para obter as informações do Arduino a cada intervalo de tempo, foi configurado o período de atualização no ScadaBR em um valor de 10 segundos para cada solicitação de informações do supervisor. Também, a fim de estabelecer a comunicação com o microcontrolador, foram definidos o endereço IP e a porta a qual o controlador está fornecendo as informações da residência. O tempo de tentativa (timeout) e a quantidade de tentativas de requisição de dados Modbus ao microcontrolador foram configurados em 500ms e 2 tentativas, respectivamente. Outros parâmetros não foram alterados, ficando com os valores padrões do ScadaBR. Na figura 8 é possível visualizar as configurações.



Propriedades do modbus IP

Nome: ARDUINO_2560

Export ID (XID): DS_527533

Período de atualização: 10 segundo(s)

Quantificação:

Timeout (ms): 500

Tentativas: 2

Apenas quantidades contínuas:

Criar pontos de monitor de escravo:

Máxima contagem de leitura de bits: 2000

Máxima contagem de leitura de registradores: 125

Máxima contagem de escrita de registradores: 120

Tipo de transporte: TCP com manter-vivo

Host: 192.168.15.100

Porta: 502

Encapsulado:

FIGURA 8 – CONFIGURAÇÃO DA REDE MODBUS NO SCADABR PARA CONEXÃO COM O ARDUINO.

FONTE: Desenvolvido pelos autores

Uma vez que o supervisório se conecta com o Arduino, o endereço Modbus de cada ponto fornecido pelo Arduino e o tipo de ponto são informados para o supervisório, como por exemplo, a luz do quarto (do tipo booleano com o endereço 03) e o sensor de temperatura (do tipo numérico inteiro com o endereço 00). Dessa forma, é possível obter os status dos pontos de cada local monitorado e controlado, conforme a figura 9.

ARDUINO_2560 - AM2560_QUARTO_FAN	0 %	22:45:00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ARDUINO_2560 - AM2560_QUARTO_FAN_STS	Desligado	22:45:00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ARDUINO_2560 - AM2560_QUARTO_LUZ_STS	Ligada	22:45:00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ARDUINO_2560 - AM2560_SALA_LUZ_STS	Desligada	22:45:00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ARDUINO_2560 - AM2560_SALA_TEMP	0 °C	22:45:00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
ARDUINO_2560 - AM2560_SALA_UMID	0 %	22:45:00	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

FIGURA 9 – PONTOS NO SCADABR FORNECIDOS PELO DISPOSITIVO ARDUINO.
FONTE: DESENVOLVIDO PELOS AUTORES

Além disso, no mesmo supervisório é possível realizar comandos de voz e texto usando a aplicação em HTML do Dialogflow (conforme figura 10). O fluxograma de comunicação do ScadaBR e Google Assistente com o Dialogflow é mostrado na figura 11.

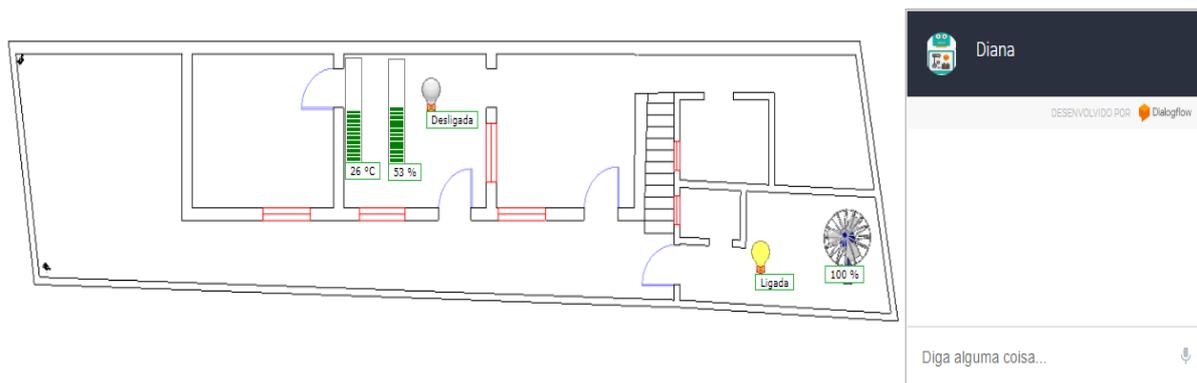


FIGURA 10 – SUPERVISÓRIO NO SCADABR.

FONTE: Desenvolvido pelos autores

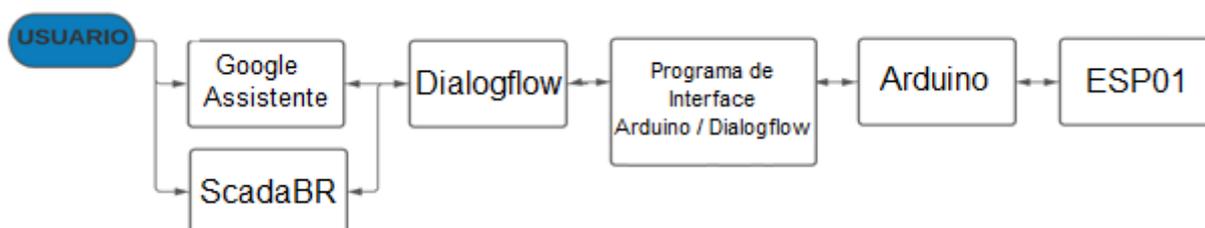


FIGURA 11 – FLUXOGRAMA DA COMUNICAÇÃO DO GOOGLE ASSISTENTE E SCADABR COM A CASA.

FONTE: DESENVOLVIDO PELOS AUTORES

Para medir a eficiência do sistema desenvolvido, foi feita uma análise quantitativa a partir de testes realizados em cada uma das interfaces disponíveis ao usuário (microfone local, supervisório ScadaBR, Google Assistente e Whatsapp Web). Os testes consistiram em solicitações por voz e por texto, realizadas para o acionamento de luz e obtenção da informação sobre temperatura e umidade do cômodo, sendo utilizadas variações de frases para os acionamentos (exemplo: “ligue a luz do quarto” / “acenda a luz do quarto”, etc.).

As duas variáveis analisadas foram o tempo de resposta (tempo entre o comando e o acionamento) e a taxa de acerto do sistema (porcentagem de comandos entendidos em relação aos comandos efetuados). Os resultados são mostrados respectivamente nas Tabelas 2 e 3.

Tabela 3 – Análise do tempo de resposta das solicitações.

Tempo de resposta			
Plataforma	Média (segundos / solicitação)	Nº de Solicitações	Tempo total (segundos)
GoogleAssistente	0,795	39	30,986
ScadaBR	1,484	36	53,426
WhatsappWeb	5,235	35	183,218
microfone_local	1,990	43	85,555
Total	2,308	153	353,185

FONTE: Elaborada pelos autores

Tabela 4 – Análise da eficiência das solicitações.

Eficiência das solicitações				
Plataforma	Entendidas	Não entendidas	Total	Taxa de acerto (%)
GoogleAssistente	38	1	39	97,44
ScadaBR	36	0	36	100,00
WhatsappWeb	30	5	35	85,71
microfone_local	25	2	27	92,59
Total	129	8	137	94,16

FONTE: Elaborada pelos autores

Fazendo a análise dos dados, é notória a eficiência no tempo de resposta da plataforma Google Assistente e ScadaBR. Pelo fato da API Dialogflow ser propriedade da Google e do ScadaBR utilizar código da API embutido via HTML, o Google Assistente e o supervisor são integrados diretamente à agente Diana, resultando em taxas de acerto maiores do que as apresentadas plataformas que necessitam de uma aplicação intermediadora em Python, como nos casos do WhatsappWeb e microfone_local.

Na Tabela 5 são mostrados detalhes do meio de comando (texto ou áudio) para a plataforma Whatsapp Web.

Tabela 5: análise do Whatsapp

Especificação das solicitações				
Plataforma	Entendidas	Não entendido	Total	Taxa de acerto (%)
WhatsappWeb Audio	11	5	16	68,75
WhatsappWeb Texto	19	0	19	100,00

FONTE: Elaborada pelos autores

A interface do Whatsapp Web foi a que mais se desviou das taxas de acertos das outras interfaces. Em uma análise mais detalhada de seus testes, é possível determinar que as dificuldades são encontradas apenas no entendimento de solicitações de áudio. Isso acontece, pois, o modo utilizado para fazer a integração entre a plataforma Whatsapp e a aplicação propriamente desenvolvida (utilizando a biblioteca Selenium) não é tão eficiente se comparado com a API própria do Whatsapp.

Sob análise qualitativa, o sistema foi capaz de reconhecer os comandos de voz e texto com variações e integrar os dispositivos via Internet. Na tabela 6, são vistas as características do objeto desta em comparação com o trabalho de Goll.

Tabela 6 – Comparativo qualitativo com o sistema VIKI de Goll

Comparativo de características		
Característica	Diana - DialogFlow	VIKI
Reconhecimento de variações de frases / voz livre	x	
Reconhecimento e sínteses do idioma português brasileiro	x	x
Síntese de voz	x	x
Capacidade de diálogo básico	x	x
Integração com Internet das Coisas	x	x

FONTE: Elaborada pelos autores

CONSIDERAÇÕES FINAIS

Com o propósito de simplificar ações do dia a dia de moradores, um sistema de automação residencial com reconhecimento de fala, utilizando uma agente da Dialogflow (Diana), foi apresentado neste artigo. Para que isso fosse

possível, foi implementado um programa em Python capaz de integrar o funcionamento de microcontroladores e dispositivos IoT com a API de entendimento de linguagem natural. Além disso, foram desenvolvidas interfaces de comunicação em plataformas que podem ser familiares aos usuários e que proporcionam a supervisão da automação.

Nos testes realizados, o sistema se mostrou eficiente, já que consegue atender o pedido do usuário em um tempo curto, comparando a média do tempo de resposta de 2,308 segundos com o tempo de resposta obtido por Neto (2018), de 4 segundos. Quanto à média da taxa de acerto (94,16%), o projeto também pode ser considerado competente na interpretação das intenções do usuário a partir do reconhecimento de texto e voz. De modo geral, as quatro formas de comunicação se mostram eficientes tanto no tempo de resposta quanto nas solicitações entendidas pela Diana.

REFERÊNCIAS BIBLIOGRÁFICAS

BOHOUTA, G.; KĚPUSKA, V. Comparing Speech Recognition System (Microsoft API, Google API and CMU Sphinx) International Journal of Engineering Research and Application ISSN: 2248-9622, Vol. 7, Issue 3, (Part -2). 2017. pp.20-24. Disponível em: <[http://www.ijera.com/pages/v7no3\(v2\).html](http://www.ijera.com/pages/v7no3(v2).html)>. Acessado em 10 de junho de 2020.

FREITAS, P. T. P. Sistema de automação residencial para deficientes visuais baseado em reconhecimento de voz. Monografia (Graduação em Engenharia Elétrica) – Universidade Federal do Ceará, Fortaleza. 2019. 78 f. Disponível em: <<http://www.repositorio.ufc.br/handle/riufc/49960>>. Acessado em 10 de junho de 2020.

GOLL, L. H. Viki: Inteligência Virtual com interação por voz aplicada à Internet das Coisas. Monografia (Graduação em Ciência da Computação) – Faculdade Regional de Blumenau, Blumenau. 2016. 58 f. Disponível em: <<http://dsc.inf.furb.br/tcc/index.php?cd=6&tcc=1798>>. Acessado em 10 de junho de 2020.

NETO, L. D. Protótipo de automação residencial utilizando uma assistente de voz. Monografia (Graduação em Ciência da Computação) – Universidade Regional de Blumenau, Blumenau. 2018. 61 f. Disponível em: <<http://dsc.inf.furb.br/tcc/index.php?cd=6&tcc=1910>>. Acessado em 10 de junho de 2020.