Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

MARIANA - ASSISTENTE VIRTUAL PERSONALIZADO PARA LINUX DESKTOP

MARIANA - CUSTOM VIRTUAL ASSISTENT FOR LINUX DESKTOP

Luiz Felipe Corazza Sauer[1] José Rafael Pilan[2]

RESUMO

A tecnologia de assistentes virtuais, que frequentemente é utilizada como agente de automação, vem ganhando espaço devido às suas vantagens industriais e domésticas. Contudo, embora esse recurso tenha potencial para ocupar boa parte do mercado, algumas pessoas resistem em conhecer e usar seus benefícios devido ao preço e resistência a alta modernidade. Outra limitação que ocorre atualmente é a quantidade reduzida de assistentes disponíveis para utilização. O presente trabalho tem como objetivo o desenvolvimento de um Assistente Virtual controlado por voz que atua em desktops Linux e fornece recursos suficientes para o usuário exercer um menor esforço para utilizar o computador. O projeto possui código aberto facilitando assim que estudantes, desenvolvedores e usuários possam eventualmente realizar melhorias ou criar o seu próprio assistente com funções e ações personalizadas. O desenvolvimento foi feito a partir de uma linguagem de programação de alto nível (Python) e com foco em programação voltada para recursos internos do sistema operacional, também foi elaborado uma comunicação com partes elétricas de uma residência pela plataforma Arduino permitindo assim automatizar processos simples.

Palavras-chave: Automação, assistente, reconhecimento de voz.

ABSTRACT

Virtual assistant technologies, which are frequently used as automation agents, have been growing because of its industrial and domestic advantages. Although such resource is strong enough to rule a good part of the market, some people still resist getting to know about them and their benefits due to their prices and resistance to high modernity. Other limitation is the reduced number of available assistants, which can be used. This paper aims at developing a voice-controlled virtual assistant that works for Linux desktops and provides enough resources for the user to make the least effort to use their computer. This project has open code/source, which makes it easier for students, developers and users to either improve or create their own assistants with personalized functions and actions. Its development happened through a high-level programming language (Python) and it was focused on programming for internal operating system resources. It was also elaborated a type of communication with the electrical parts of a home on Arduino platform which made it possible to make simple automatic processes.

Key Words: Automation. assistent. speech recognition.

[1]Graduado no curso Análise e Desenvolvimento de Sistemas (FATEC, Botucatu-SP), email: luisguilhermedemello@gmail.com
[2] Docente na Faculdades Integradas de Botucatu (Unifac)

Recebido em setembro de 2019 e Aceito em fevereiro de 2021.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

1. INTRODUÇÃO

É difícil pensar uma situação ou contexto na qual a tecnologia não está presente: medicina, engenharia, escolas, indústria e segurança, todas são áreas onde podemos observar uma marcante expressão tecnológica. Ao longo do tempo, tornou-se comum pensar na tecnologia como sendo um recurso no qual apenas parte da população pode desfrutar de sua essência e criações, devido a alto custo e resistência por parte dos usuários finais, porém, a tecnologia disseminou-se a tal ponto que hoje tornou-se mais acessível e está ao alcance de todos (TEZA, 2002).

Criações tecnológicas podem gerar automatização de máquinas em áreas industriais, como por exemplo robôs, que substituem trabalho braçal humano; facilidade de acesso a sistemas e recursos educacionais, plataformas de auxílio ao conhecimento; otimização de desempenho relacionado a saúde, equipamentos hospitalares; segurança de áreas empresariais e domésticas, câmeras e sistemas de alarmes.

Assim como as tecnologias citadas, o conceito de Assistente Virtual (AV) se concretizou com o intuito de facilitar ações humanas rotineiras e proporcionar conforto e segurança para os usuários. Os AV's possuem a capacidade de reconhecer comandos de voz e podem executar ações usando o reconhecimento como entrada do sistema, pode responder algumas perguntas, como uma solicitação da previsão do tempo para o dia atual ou executar ações que controlam partes elétricas de uma casa e manipular eventos de um computador pessoal ou celular.

Alguns AV's, além de proporcionar facilidade e rapidez em tarefas dentro de um sistema computacional, possuem a capacidade de se comunicar com o ambiente externo ao sistema, como circuitos elétricos de equipamentos eletrônicos. Tendo em vista o aumento de segurança, conforto e praticidade, a automatização de recursos residenciais, também chamada de domótica, diz respeito a integração de sistemas inteligentes com uma residência (ACCARDI e DODONOV, 2012).

A automação residencial tem mostrado que a integração de dispositivos eletroeletrônicos e eletromecânicos aumenta consideravelmente os benefícios se comparados com os sistemas isolados, de eficiência limitada, também é uma aliada na redução do consumo de recursos como água e energia elétrica, além de trazer maior conforto e segurança ao usuário (BOLZANI, 2007).

Cada um dos assistentes virtuais (Alexa, Siri, Cortana...) possuem caráter único e por mais que suas funções sejam de grande importância e ajuda, são limitadas às ações que podem fornecer, isto é, assistentes possuem uma programação fixa que geralmente é limitada à

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

liga/desliga, abre/fecha, pesquisas simples e outras funcionalidades específicas de cada assistente, a liberdade de modificação de ações para implementar novas funcionalidades é praticamente nula.

Normalmente, os assistentes pessoais domésticos têm como público-alvo pessoas que queiram se integrar com a tecnologia e seus avanços, porém, nem sempre seus usuários sentem a necessidade de utilizá-lo, seja por já estarem acostumados a digitar suas pesquisas ou por falta de utilidade. Tal tecnologia pode sim ser vista como algo benéfico e com uma visão futurista e sadia, porém, dessa forma, as tarefas automatizadas se tornam extremamente monótonas e sem grande importância, impossibilitando o uso para coisas mais voltadas a trabalho, estudos e desenvolvimento intelectual, fazendo com que o assistente seja visto como algo a ser usado apenas por diversão e não como uma vantagem diária.

As pessoas se interessam por esse tipo de tecnologia, porém os assistentes não são voltados a ações mais importantes do que checar a previsão do tempo ou ver o placar de um jogo de futebol pelo Google, fazendo com que boa parte das pessoas deixem de usá-los. Isso acontece não porque os assistentes são insuficientes, mas porque as tarefas que eles realizam, na maioria das vezes, são muito simples e limitadas, fazendo com que o usuário opte por realizá-las manualmente.

O presente trabalho tem como objetivo o desenvolvimento de um Assistente Virtual, nomeado de Mariana, personalizado e controlado por voz que atua em sistemas operacionais Linux baseados em Ubuntu, realizando tarefas rotineiras como fazer pesquisas, abrir sites, e algumas mais complexas como reconhecer e replicar frases faladas de longa metragem em *softwares* de escrita (*Microsoft Word, LibreOffice, Gedit*, entre outros), reconhecer nomes de sites e recursos dos mesmos para abri-los com pequenas *triggers* de fácil entendimento, manipular janelas abertas (minimizar, maximizar, ocultar, fechar e trocar de janela), navegador, YouTube, Google, volume e monitoramento de recursos do sistema. Além da parte relacionada ao controle do ambiente interno do sistema operacional, isto é, com os recursos que o mesmo oferece como navegador, editores de texto e visualizadores de imagem, o projeto Mariana também visa ser capaz de controlar os sistemas elétricos da residência do usuário através da plataforma Arduino, onde a comunicação se dará via Bluetooth.

2 MATERIAL E MÉTODOS

Em seguida, serão descritas as ferramentas de desenvolvimento utilizadas para a realização do projeto.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

2.1 Hardware

Para o desenvolvimento do projeto, foi utilizado um notebook QBEX com as seguintes especificações:

Quadro 1 - Configurações do notebook utilizado

Sistema operacional:	Ubuntu 18.04.2 LTS 64 bits
Memória RAM:	3,8 GB
Processador:	Intel Core i5-2410M CPU @ 2.30GHz x 4
Espaço interno:	Disco Rígido 500GB

Fonte: O autor, 2019.

A parte responsável pela automação externa ao sistema é efetuada a partir da plataforma Arduino, um *mini*computador com a possibilidade de programar processos de entrada e saída entre o dispositivo e componentes/módulos externos conectados a ele (MCROBERTS, 2015). O protótipo foi montado contendo os seguintes módulos e utilitários:

- 1 Arduino Uno (Plataforma para prototipagem)
- 1 Módulo Relé (Interruptor digital)
- 1 Módulo Bluetooth (Dispositivo de envio e recebimento de informações)
- 1 Protoboard Mini (Placa de prototipagem)
- Cabos de conexão
- 2 Baterias de Lítio de 3,3V
- 1 Acoplador de bateria de lítio

2.2 *PyCharm*

PyCharm, desenvolvido pela JetBrains em 2010, é um editor de texto que embora seu foco seja fornecer ao programador recursos que facilitam a escrita de um código em Python, também dá suporte a digitação de linguagens como *Django*, *Flask, Pyramid*, HTML, CSS, JavaScript e XML (CAELUM, 2019).

O editor fornece suporte à instalação e remoção de módulos externos de forma nativa e disponibiliza diferentes interpretadores de códigos que podem ser escolhidos pelo usuário.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

O *PyCharm* foi escolhido para a realização deste projeto em virtude da possibilidade de redução do tempo total de desenvolvimento, por facilitar a indentação e fornecer sugestões para completar as linhas de comando, bem como conseguir fazer a análise completa do código.

2.3 Google Chrome

Embora necessite usar mais recursos do computador, como memória RAM, Disco Rígido e cache do sistema, o browser Chrome, desenvolvido pela empresa norte americana Google, se mostra melhor em desempenho e mais rápido, se comparado a outros navegadores como *Firefox* e Opera (COSTA, 2015) e, por isso, é utilizado como principal interface para processos de automação de recursos WEB no projeto Mariana.

2.4 Python

Python é uma linguagem de alto nível que pode ser escrita tanto na forma estruturada quanto orientada a objeto, sua sintaxe possui maior clareza e suas linhas de comando são independentes de ponto e vírgula (;), como em outras linguagens, favorecendo a legibilidade do código fonte e tornando-a mais produtiva (SANNER, 1999).

A linguagem inclui diversas estruturas de alto nível, incluindo listas e dicionários, e uma vasta coleção de módulos prontos para uso, além de *frameworks* de terceiros que podem ser adicionados (BORGES, 2018).

Python é utilizado no desenvolvimento de sistemas e se mostra eficiente como linguagem de script e prototipagem, permitindo a automatização de tarefas e a criação de programas de comunicação com microcontroladores.

As seguintes bibliotecas foram utilizadas em conjunto com o *Python* para execução do projeto:

2.4.1 *Speech Recognition*

A tecnologia de reconhecimento de voz é possível graças a interação e avanços do computador e placas de som, tornando real a possibilidade de converter som em dados digitais (VALIATI, 2000).

O Projeto Mariana possui o intuito de ser um Assistente Virtual que aceite a fala informal do cotidiano. Para que isso fosse possível, utilizou-se a biblioteca *Speech Recognition*,

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

que permite que o desenvolvedor escolha e faça uso de uma API em nuvem (Uma API (*Application Programming Interface*), do português, Interface de Programação de Aplicações, é uma biblioteca que permite que o desenvolvedor utilize recursos de sistemas fora de seu ambiente de trabalho).

No caso da biblioteca *Speech Recognition,* os recursos que são consumidos pelas API's são os serviços de reconhecimento de voz de empresas que disponibilizaram seus processos para desenvolvedores utilizarem. As possibilidades de API's que a biblioteca fornece são:

- Google *Speech Recognition;*
- Wit.ai;
- *Microsoft Bing Voice Recognition;*
- *Houndify* API;
- IBM *Speech to Text.*

A API usada nesse projeto é a da Google, devido à alta precisão no reconhecimento da fala e suporte a diferentes línguas ao mesmo tempo (KËPUSKA, 2017). Por ser um projeto que usa uma ferramenta que utiliza API's de serviços na nuvem para reconhecimento de fala, a conexão com a Internet é crucial para o seu funcionamento.

A biblioteca faz uso do módulo *PyAudio* para acessar o microfone e então fazer o reconhecimento, portanto, a instalação do mesmo é essencial.

Após a solicitação de reconhecimento de voz, a biblioteca retorna um valor em texto que deverá ser atribuído a uma variável do script para posteriormente ser submetida a testes e condicionais que farão com que o Assistente Virtual Mariana realize suas ações.

2.4.2 Selenium

A biblioteca Selenium consiste em uma ferramenta que consegue automatizar ações em um navegador (*Browser*), como clicar e extrair informações de elementos HTML, acessar sites, voltar, avançar, recarregar e preencher formulários e campos (RAGHAVENDRA, 2021).

A sua utilização requer o *driver* referente ao navegador que está sendo utilizado para os testes automatizados, que pode ser baixado no site do fabricante do mesmo.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

2.4.3 *PyAutoGUI*

O projeto Mariana é um Assistente Virtual desenvolvido como uma automação dentro de um sistema computacional, sendo uma de suas principais funcionalidades a escrita automática de uma frase falada, e com isso existe a necessidade da utilização de um recurso que supra a necessidade de escrever sem o teclado. Para que isso seja possível, foi utilizada a biblioteca *PyAutoGUI*.

Seu funcionamento consiste em automatizar os sinais de entrada do mouse e teclado do computador. Ele possibilita que, através de linhas de comando em *Python*, o desenvolvedor simule movimentos e cliques dos botões do *mouse*, pressionamento (inclusive manter pressionado e soltar) e combinações de pressionamento de teclas do teclado (PYAUTOGUI, 2014).

Além da escrita automática em editores de texto, seus recursos serão de extrema importância para que o usuário alterne (Alt + Tab), diminua (*Windows* + seta para baixo), maximize (*Windows* + seta para cima), oculte (*Windows* + H) e feche (Alt + F4) janelas abertas em seu sistema, a partir de comandos de voz.

2.4.4 PySerial

Para a comunicação da Mariana com partes elétricas da casa, foi necessária a utilização de uma biblioteca capaz de estabelecer uma conexão com o Arduino via serial, enviando dados um bit de cada vez, sequencialmente, num canal de comunicação, como *Bluetooth*, ou barramento, via USB. A biblioteca *PySerial* atende essa necessidade por conseguir acessar uma porta de comunicação do computador com um dispositivo pareado via *Bluetooth* (PYSERIAL, 2021).

Através de apenas uma linha de comando, a biblioteca estabelece uma ligação entre o computador, onde a Mariana se encontra, e um módulo *Bluetooth* conectado ao Arduino, sendo possível o envio e recebimento de informações de maneira rápida e segura.

2.5 Metodologia de reconhecimento de voz

As estruturas de acionamento de funções foram feitas com o método "*contains*", em português, "contém". Esse método, ao invés de verificar a equidade de duas estruturas de texto (*String*) e entrar no laço condicional, se as mesmas forem exatamente iguais, verifica se

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

possuem, em alguma parte de seu escopo, sequências de texto compatíveis uma com a outra. Isso garante que a Mariana verifique e entenda o que o usuário quer executar, mesmo dito informalmente e fora de ordem. A Figura 1 mostra um exemplo do comportamento do Assistente Virtual Mariana, quando submetida ao reconhecimento de frases formais e informais:

Figura 1 - Teste de reconhecimento de voz de duas maneiras distintas

```
Diga algo!
Mariana reconheceu: mariana abrir navegador
Abrindo o navegador...

Diga algo!
Mariana reconheceu: mariana eu preciso que você abra nesse momento o navegador
Abrindo o navegador...
```

Fonte: O autor, 2019.

Mariana, inicialmente, esperou que o usuário iniciasse sua fala e, logo em seguida, em ambas as ocasiões, transformou a frase em texto puro, reconhecendo as *triggers* “abrir”, “abra” e “navegador”.

Para fornecer um maior conforto para o usuário, Mariana tem a capacidade de reconhecer um *site* através do seu nome principal (por exemplo, YouTube”) e não pelo seu endereço completo (https://www.youtube.com.br). Para que o usuário não seja obrigado a falar o endereço completo do site a ser acessado toda vez que sentir necessidade de abrir o mesmo, o que pode ser cansativo, Mariana, através de uma *trigger* de voz, acessa o atual endereço em que o usuário está conectado e o guarda em um arquivo de texto específico para *triggers* de sites dentro da pasta do projeto. O nome, que será utilizado como *trigger*, será o texto que estiver entre os dois primeiros pontos (.) do endereço completo.

Toda vez que o script for iniciado, uma estrutura de dicionário será carregada e ficará à disposição do usuário durante seu tempo de execução.

Dessa forma, ao fazer o reconhecimento de voz e Mariana detectar a palavra “abrir” na *string* retornada pela biblioteca de reconhecimento de voz, um laço de repetição será executado para cada item da estrutura e irá verificar se a fala contém o nome de algum site incluso no dicionário de sites. Caso sim, Mariana irá abri-lo. Esta função suporta a inicialização de vários sites ao mesmo tempo.

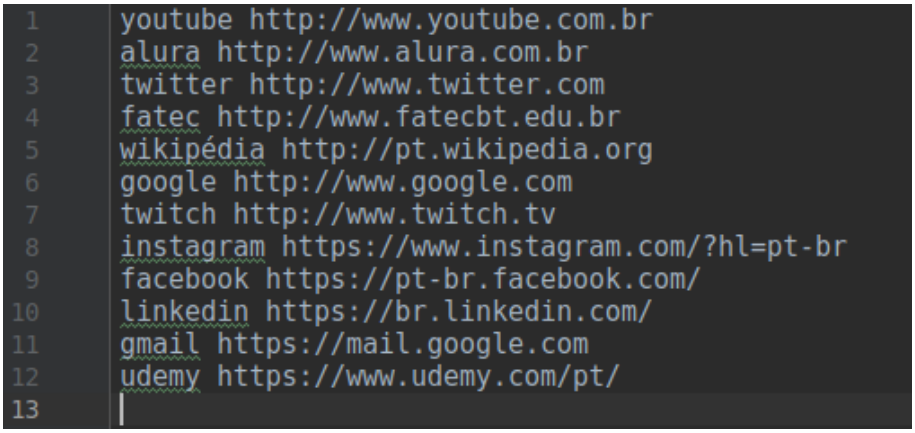
Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

2.6 Automatização do *Browser*

Para que o projeto reconhecesse recursos WEB com mais facilidade e rapidez, foram criadas duas estruturas de dicionário responsáveis por gerenciar nomes de sites. Essas estruturas computacionais têm a capacidade de armazenar dois valores: um apelido e uma referência. Em casos onde a referência pode ser um valor complexo e difícil de ser digitado, um dicionário pode ser implementado para facilitar o uso dos mesmos.

A primeira estrutura, cujo nome foi dado "*web_dict*", será preenchida com informações contidas em um arquivo de texto dentro do projeto, onde cada linha contém a *trigger* do site e o seu endereço completo. A Figura 2 mostra os dados do arquivo:

Figura 2 - Arquivo de sites



Fonte: O autor, 2019.

Dessa forma, ao iniciar a execução do script principal, cada linha será alocada como uma ocorrência no dicionário "*web_dict*", resultando na chamada de abertura do site apenas com seu nome.

A segunda estrutura, cujo nome foi dado de "*web_search*", será preenchida com as informações contidas em um arquivo de texto interno existente no projeto, contendo o nome do site, bem como seu endereço completo de pesquisas, isto é, seu recurso de procura interno.

Para que as pesquisas do usuário em sites como Google e YouTube fossem efetuadas de maneira mais rápida, foram criadas triggers que modificam o URL de pesquisa de um determinado site que o usuário deseja procurar. A Figura 3 mostra o conteúdo do arquivo de texto contendo as informações de endereço de pesquisa de alguns sites:

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

Figura 3– Arquivo de endereços de pesquisa

```
1  google http://www.google.com/search?q=
2  youtube http://www.youtube.com/results?search_query=
3  twitch http://www.twitch.tv/
4  wikipédia http://pt.wikipedia.org/wiki/
5
```

Fonte: O autor, 2019.

Quando solicitado, Mariana fará uma busca automática no site que o usuário solicitar, verificando o endereço de pesquisa na estrutura de dicionário anteriormente criada e acrescentando ao seu final, um texto criado com uma segunda chamada de reconhecimento de voz.

3 RESULTADOS E DISCUSSÃO

O Assistente Virtual Mariana, ao analisar as palavras faladas pelo usuário compara com as palavras chaves existentes na biblioteca, reconhece as *triggers* de abertura (abre, abra) juntamente com o site a ser aberto especificado pelo usuário (FIGURA 4). Na Figura 5 é demonstrado o resultado de uma busca no site Google, tendo como resultado a exibição da página através do navegador (FIGURA 6).

Figura 4 – Reconhecimento de sites através do seu nome

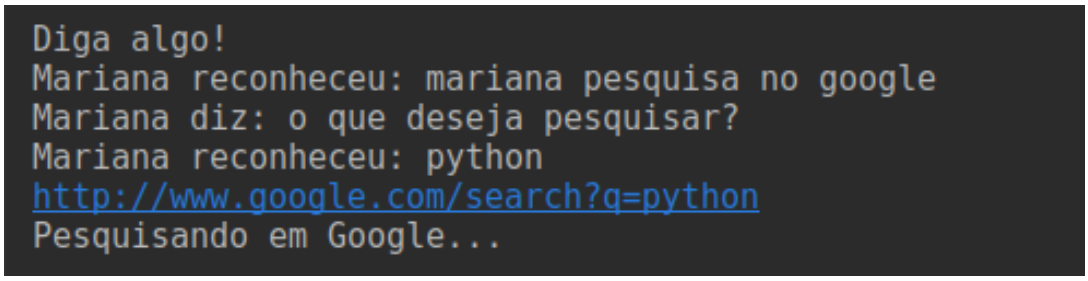
```
Diga algo!
Mariana reconheceu: mariana abra o youtube
http://www.youtube.com.br
Abrindo Youtube...

Diga algo!
Mariana reconheceu: mariana abre o facebook para mim por favor
https://pt-br.facebook.com/
Abrindo Facebook...
```

Fonte: O autor, 2019.

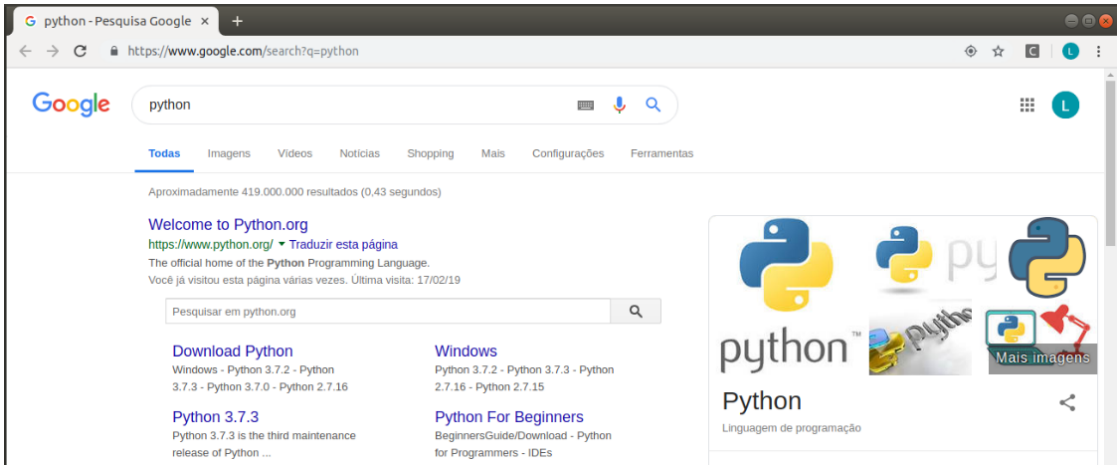
Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

Figura 5 – Resultado de solicitação de pesquisa



Fonte: O autor, 2019.

Figura 6 – Resultado de solicitação de pesquisa com a abertura da url pelo navegador



Fonte: O autor, 2019

Outras funcionalidades implementadas para atuar em conjunto com o navegador:

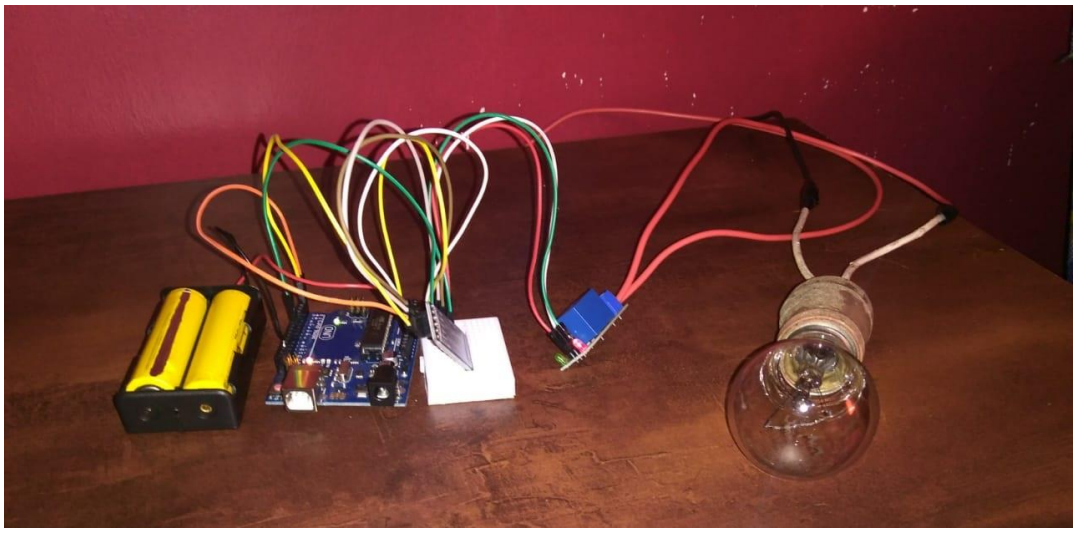
- Pausar e restaurar vídeos no YouTube, bem como abrir em modo teatro e tela cheia;
- Voltar, avançar e recarregar página;
- Trocar e fechar abas abertas;
- Reconhecer o site aberto atual e fazer pesquisas, caso o usuário solicite, sem abrir uma nova aba;
- Aprender o site atual automaticamente com a trigger “Mariana, aprenda esse site”, escrevendo-o no final do arquivo de URL’s;
- Subir e descer a barra de rolagem do navegador.

Para que Mariana fosse capaz de controlar um circuito elétrico fora de seu ambiente de atuação, isto é, fora do sistema operacional, foi montado um protótipo com a plataforma

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

Arduino, que visa receber informações do script do projeto e fazer suas atividades de acordo com as mesmas. O protótipo pode ser visto na Figura 7:

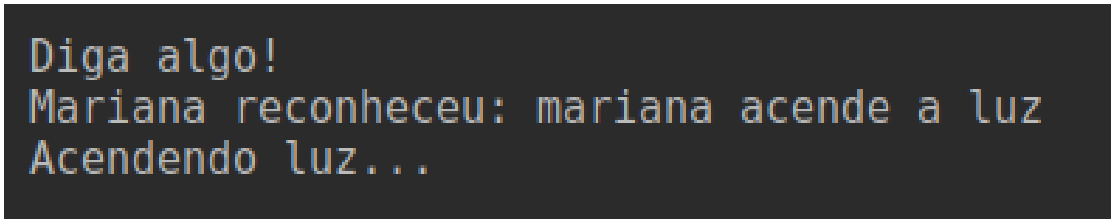
Figura 3 – Protótipo com Arduino



Fonte: O autor, 2019.

Quando acionada a trigger de acender/apagar a luz, Mariana envia um código via Bluetooth para o Arduino, que por sua vez, os recebe, lê e verifica em suas linhas de comando se o usuário está solicitando que a luz acenda ou apague. As Figuras 8 e 9 mostram tal função na prática:

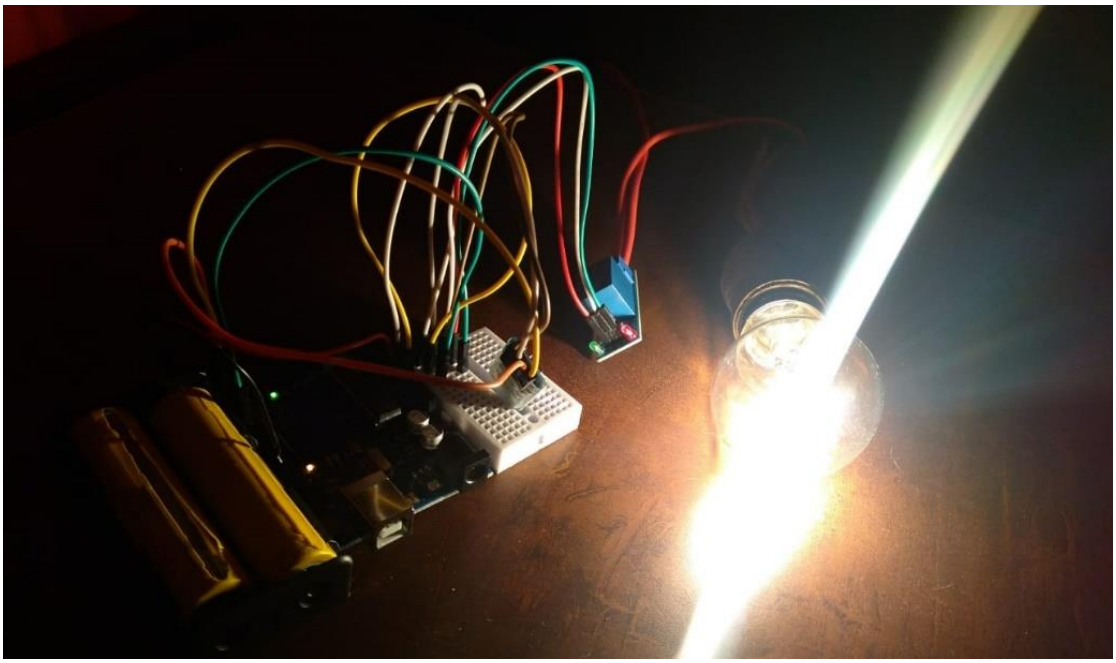
Figura 4 - Resultado da solicitação para ligar a luz



Fonte: O autor, 2019.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

Figura 5 - Luz acesa após solicitação via AV



Fonte: O autor, 2019.

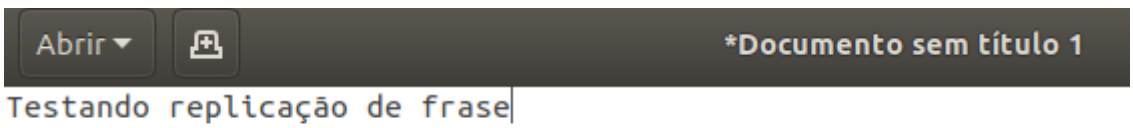
Para que o usuário seja capaz de fazer grande parte de suas ações no sistema apenas com comando de voz, foi desenvolvida uma *trigger* que é capaz de replicar frases faladas de longa metragem em qualquer programa que aceite texto. As Figuras 10 e 11 mostram o seu funcionamento:

Figura 6 - Solicitação de replicação de frase falada

```
Diga algo!
Mariana reconheceu: mariana escreva minhas palavras
Oque deseja escrever?
Mariana reconheceu: testando replicação de frase
```

Fonte: O autor, 2019.

Figura 7 - Resultado da replicação da frase



Fonte: O autor, 2019.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

Outras funcionalidades implementadas:

- Maximizar, minimizar, fechar e ocultar janelas abertas;
- Controle de volume do sistema operacional;
- Abertura de aplicativos como Terminal, *Notepad* e *LibreOfice*;

4 CONCLUSÕES

O Assistente Virtual mostrou-se eficiente ao realizar as diferentes tarefas e ações que lhe foram programadas. Mariana consegue ser uma solução eficaz e inovadora para quem deseja se integrar com a tecnologia de assistentes virtuais por ser um AV que permite sua personalização através da alteração da programação no código fonte. Fazer com que o AV suprisse a necessidade de entender as falas informais do cotidiano e reconhecer *triggers* de uma maneira dinâmica, para um maior conforto do usuário, assim como o processo e tratamento das frases em texto foi possível pela utilização da biblioteca *Speech Recognition*. Ao invés do usuário ser obrigado a falar exatamente o que foi estabelecido previamente como *trigger*, é possível acionar as funções da Mariana com falas longas e informais, desde que, na frase, contenha as palavras chaves. Para que o projeto seja ainda mais eficaz e útil, no futuro serão desenvolvidas novas formas de interação com o ambiente externo ao sistema junto com o Arduino, automatizando equipamentos eletrônicos e manipulando dados que a plataforma irá enviar ao computador como temperatura e umidade.

REFERÊNCIAS

ACCARDI, A.; DODONOV, E. Automação residencial: elementos básicos, arquiteturas, setores, aplicações e protocolos. **Revista TIS**, v. 1, n. 2, 2012. Disponível em <http://revistatis.dc.ufscar.br/index.php/revista/article/view/27/30>. Acesso: 05 jun. 2019.

BOLZANI, C. Desmistificando a domótica.**Escola Politécnica da Universidade de São Paulo**, 2007. Disponível em <http://www.bolzani.com.br/artigos/art01_07.pdf>. Acesso: 28 mai. 2019.

BORGES, L. E. **Python para desenvolvedores: aborda Python 3.3**. Novatec Editora, 2014.

CAELUM. **Python e Orientação a Objetos**. Disponível em <https://www.caelum.com.br/apostila-python-orientacao-objetos/>. Acesso: 05 jun. 2019.

COSTA, J. **Os melhores navegadores web para Linux**. 2015. Disponível em <https://www.linuxdescomplicado.com.br/2015/09/os-melhores-navegadores-web-para-linux.html> Acesso: 01 jun. 2019.

Tekhne e Logos, Botucatu, SP, v.12, n.3, dezembro, 2021.
ISSN 2176 – 4808

KËPUSKA, V.; BOHOUTA, G. Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx**). Int. J. Eng. Res. Appl**, v. 7, n. 03, p. 20-24, 2017.

MCROBERTS, M. (2015). Arduino básico. **Novatec Editora, 2ªed**. Disponível em <https://s3.novatec.com.br/capitulos/capitulo-9788575224045.pdf>. Acesso: 25 jun. 2019.

PYAUTOGUI. Documentação de biblioteca Python. 2014. Disponível em <https://pyautogui.readthedocs.io/en/latest/>. Acesso: 06 jun. 2019.

PYSERIAL. Disponível em < https://pyserial.readthedocs.io/en/latest/pyserial.html >. Acesso: 16 fev. 2021.

SANNER, M. F. 1999. Python: a programming language for software integration and development. **J Mol Graph Model**, 17(1), 57-61. Disponível em <https://pdfs.semanticscholar.org/409d/3f740518eafcfaadb054d9239009f3f34600.pdf>. Acesso: 06 jun. 2019.

RAGHAVENDRA, S. Introduction to Selenium. **Python Testing with Selenium**. Apress, Berkeley, CA, 2021. p. 1-14.

TEZA, V. R. (2002). Alguns aspectos sobre a automação residencial: domótica. Disponível em <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/83015/212312.pdf?sequence=1&isAllowed=y>. Acesso: 01 jun. 2019.

VALIATI, J. F. (2000). **Reconhecimento de voz para comandos de direcionamento por meio de redes neurais**. Disponível em <https://www.lume.ufrgs.br/handle/10183/2947>. Acesso: 05 jun. 2019.