FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS – FATECS
CURSO

RENATO DE SOUSA BARBOSA
21853879

ARQUITETURA PARA ASSISTENTE DE BUSCA VIRTUAL

BRASÍLIA
2022

Renato de Sousa Barbosa

ARQUITETURA PARA ASSISTENTE DE BUSCA VIRTUAL

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia Civil, Elétrica ou de Computação do UniCEUB– Centro Universitário de Brasília

Orientador (a): Fabio Oliveira Guimarães

BRASÍLIA
2022

Renato de Sousa Barbosa

ARQUITETURA PARA ASSISTENTE DE BUSCA VIRTUAL

Trabalho de Conclusão de Curso (TCC) apresentado como um dos requisitos para a conclusão do curso de Engenharia Civil, Elétrica ou de Computação do UniCEUB – Centro Universitário de Brasília

Orientador (a): Fabio Oliveira Guimarães

Brasília, 2022.

BANCA EXAMINADORA

_____

Nome e titulação.
Orientador (a)

_____

Nome e titulação.
Examinador (a)

_____

Esp. Fábio Oliveira Guimarães
Examinador (a)

Arquitetura Para Assistente de Busca Virtual
Architecture For Virtual Search Assistant

Renato de Sousa Barbosa[1], Fabio Oliveira Guimarães[2].

RESUMO

Este projeto tem como objetivo desenvolver uma arquitetura para a implementação de buscas e execução de múltiplas funções para assistentes virtuais, sendo mais precisa nos resultados das pesquisas trazendo uma nova experiência para o usuário e automatizando um ciclo de tarefas. O conteúdo encontrado através do sistema será filtrado e selecionado a partir de parâmetros condicionais que indiquem que a pesquisa foi a mais relevante. O sistema será capaz de armazenar a busca para que a mesma possa ser usada depois com a execução de uma nova tarefa, será possível automatizar várias tarefas de navegação no computador. A arquitetura é composta por reconhecimento de voz, sintetizador de voz, buscador, núcleo de funções, sistema que executa múltiplas funções, com isso o uso de uma interface gráfica pode ser reduzida, podendo ter vários comandos que de forma direta pesquisa e faz várias tarefas simultâneas.

Palavras-chave: Assistente. Buscador. Automatização. Multiplas Funções.

Abstract: This project aims to develop an architecture for the implementation of searches and execution of multiple functions for virtual assistants, being more precise in the research results, bringing a new experience to the user and automating a cycle of tasks. The content found through the system will be filtered and selected from conditional parameters that indicate that the search was the most relevant. The system will be able to store the search so that it can be used later, with the execution of a new task, it will be possible to automate various navigation tasks on the computer. The architecture is composed of speech recognition, speech synthesizer, search engine, core functions, system that performs multiple functions, with this the use of a graphical interface can be reduced, being able to have several commands that directly search and perform several tasks simultaneous.

keywords: Assistant. Search Engine. Automation. Multiple Functions.

[1] UniCEUB, aluno.
[2] UniCEUB, orientador.
[3] UniCEUB, primeiro examinador.
[4] UniCEUB, segundo examinador.

1 INTRODUÇÃO

O avanço da tecnologia proporcionou que as pessoas pudessem ter acesso ao computador podendo fazer pesquisas em todo o globo que está conectado na internet. Com isso novas tecnologias surgiram e uma delas foi a assitente virtual que comecou sendo apenas um reconhecimento de voz para os motores de busca nos anos 90 (OLHARDIGITAL, 2020).

Existem 4 assistentes mais conhecidas a Alexa, Cortana, Google Assitant e Siri, o objetivos delas é facilitar as buscas, agendar e o que a maior parte das pessoas usam para é para transformar suas moradias em casas inteligente.

Se uma assistente virtual fosse capaz de ser um sistema operacional, consegue imaginar o que as Assistentes Virtuais vão poder fazer?

Por isso o projeto terá como objetivo fazer um assistente com a arquitetura diferente das atuais, na qual ela consegue executar multiplas funções usando um algoritmo procedural que se adapta dependendo da pesquisa do usuario. Podendo aplicar filtros nas buscas, trazendo resultados de videos, imagens e até fazer downloads, tudo isso usando a voz.

A evolução das Assistentes é uma coisa inevitavél, porém é algo que a tecnologia já é capaz de fazer, então porque não mostrar que a utilidade dela pode ir desde pessoas com necessidades especiais, como aumentar a produtividade e diminuir tempo ocioso com buscas irrelevantes.

Segundo o autor Teixeira Cruz e col. (2013), fala que combinando as tecnologias existentes elas geram algo novo. E baseado nessa premissa será criado um modelo de arquitetura para um novo modelo de assistentes virtuais.

O projeto partiu do ponto que também precisa se pensar na diversidade e inclusão. A pesquisa foi baseada em livros e artigos, para que a validação fosse melhor aceita. Nos capitulos seguintes abordaremos todos os passos explicando as definições e a praticando, combinando algoritmos e novas tecnicas para demontrar com resultados, que as assistentes virtuais podem se tornar mais uteis e inovadoras.

2 REVISÃO BIBLIOGRÁFICA

2.1 Trabalhos relacionados

Entre alguns dos assuntos que serão abordados nesse estudo podemos citar uma relação entre um artigo onde uma assistente de suporte na escrita coletiva e isso traz conceitos que podem ser abordados na função classificadora onde trará novos paradigmas de execução. (Alexandra Lorandi Macedo. 2007)

Podemos ver a rapida evolução na utilização das assistentes e esse impacto tornará possivel um novo tipo de assistente virtual. (Giovanna Dourado Marchi e col. 2019)

Mesmo com as assistentes presente nos smartfones encontram se diculdades em auxiliar os estudantes nas resoluções de suas atividades. (Daiane Rose da Silva Matos, Francisco Kelsen de Oliveira. 2011)

Usando inteligia artificial podemos melhorar a precisão na tomada de decisão e escolher melhores alternativas. (Ingredy Anhaia e col. 2018)

2.2 Assistente Virtual

As assistentes virtuais tem o foco geral em facilitar o acesso do usuario com à informação. Cerca de dois bilhoes de usuarios utilizam a internet para varias atividades, sendo que no Brasil 94,2 milhões de pessoas tem acesso a internet, então podemos dizer que praticamente metade da população utiliza internet, logo muitos dados são criados e surge a necessidade de ter uma assistente virtual inteligente.

As buscas atuais consistem em pedir uma informação à assistente por comando de voz, e ela faz uma varredura no buscador retornando apenas os links das paginas, com exceção quando se trata de definições simples. Essa busca se torna ineficiente, pois não garante que a informação que o usuario busca seja encontrada. Então no caso da assistente virtual inteligente, o usuario pediria a informação e ela entenderia o que o usuario quer através de base de dados, sinonimos e varias funções de filtros.

Bell Laborarories criou o "Audrey" em 1950, o primeiro dispositivo que reconhecia voz de digitos numericos, nos anos de 60 à 80 setenças curtas de palavras e deviam ser ditar com pausas, mas os sistemas só ficaram disponiveis em 1990, o primeiro devia ser treinando antes de ser usado tendo uma precisão de 95%, o segundo não necessita de treinamento só executar e falar. Leôncio Teixeira Cruz e col. (2013)

2.3 Sistema de Tempo Real

A definição segundo Jean-Marie Farines e col. (2000, v. 1, p. 15) 'Um Sistema de Tempo Real (STR) é um sistema computacional que deve reagir a estímulos oriundos do seu ambiente em prazos específicos."

O sistema de tempo real deve ter resposta correta em um certo prazo de tempo sem ocorrer falhas. Grande parte das implementações são consideradas sistemas reativos com restrições temporais no qual o Sistema a Controlar e o Operador são considerados como o Ambiente do Sistema Computacional. Essa interação é composta por sensores e atuadores, que são a fonte da leitura de dados.

Um Sistema de Tempo Real deve ser então capaz de oferecer garantias de correção temporal para o fornecimento de todos os seus serviços que apresentem restrições temporais.

A velocidade computacional oferece um tempo de resposta curto, mas isso não garante que o processamento temporal será atendido, ja a previsibilidade é mais importante. Para o sistema ser prevsivel ele não deve sofrer variação no hardware sem cargas e falhas, antecipando todas os prazos a partir das interaçãoes com o ambinte.

A hipotese de carga se trata da carga computacional, ou seja, a quantidade de informação processada em um intervalo de tempo.

A hipotese de falhas define as frequencia e tipo de falhas que devem ser evitadas no tempo, isso acontece quando se conhece o comportamento do sistema, considerando situações de carga simultaneas às falhas.

Para garantir a previsibilidade e necessario um conjunto de fatores como arquitetura de hardware, sistema operacional e linguagens de programação.(Jean-Marie Farines e col. 2000)

2.4 Reconhecinmento de Voz

O reconhecimento de voz automatico ja está presente em diversas apicações, renhecendo desde pequenos vocabularios até a compreensão espontanea de fala. (Vladimir Fabregas Suringué de Alencar. 2005)

Podemos separar em tres sistemas de reconhecimento de voz:
Reconhecedor de palavras isoladas – usado para reconhecer vocabularios pequenos e ambientes sem ruido.

Reconhecedor de voz contínua – Consegue reconhecer sesetenças de

linguagem natural sem necessisade de pausas entre as palavras.

Reconhecedor de voz contínua – São dificeis de implementar, porque deve ser capaz de entender vicios de linguagens e as durações de palavras desconhecidas.

2.5 Sintetizador de voz

A sitese da voz busca reproduzir a fala humana através do sinal da voz, reconhecendo naturalidade e a inteligibilidade quando o objetivo é a qualidade da voz sintetica

A voz humana é produzida pelo aparelho fonador que é composto por é composto pelo diafragma, pulmões e o trato vocal. Então quando o ar sai dos pulmões os orgão se movimentam gerando vibrações com a passagem do ar, assim o espectro do sinal produz sons reconhecidos pelas pessoas.

O primeiro sintetizador foi construido em 1922 pelo cientista Stewart, que imitava o as funcionalidades dos pulmões, com essa maquina foi criado a geração estática dos sons da vogais, logo depois o VODER (Voice Operating Demonstrator), desenvolvido por Homer Dudley, em 1939, o primeiro dispositivo sintetizador elétrico. (Fabíola Pantoja Oliveira Araújo. 2015)

Dentre as aplicações existentes o sistema TTS, converte o texto em uma voz artificial tentando ser o mais natural possivel. Esse processo é dividido em duas partes o Front-end e Back-end.

O front-end recebe o texto limpo e converte simbolos em palavras para realizar a transcrição fonetica.

O back-end é o responsavel por fazer a sintese da voz, ele converte a representação linguistica através do processamento digital de sinais.

2.6 Multiplas funções

Um thread é usado para executar uma tarefa, se existir multitrheads o processo pode executar mais de uma tarefa ao mesmo tempo, possui um id de identificação, contador, conjunto de registradores e uma pilha.

As aplicações de software modernas são executadas usando multithread, são usadas para aumentar a velocidade de desempenho e tarefas intensas.

No processo tradicional um trhead só consegue atender uma solicitação por vez, isso faria o cliente esperar. A criação de um processo é demorada e usa muitos recursos , então para não gerar overhead, usamos o mesmo com multiplos threads, permitindo atender a várias solicitações concorrentes. (Silberschatz, Abraham. 2015)

3 METODOLOGIA

3.1 Analise de requisitos

3.1.1 Requisitos Funcionais

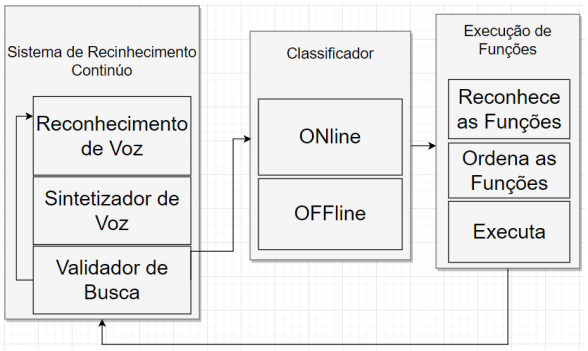
- Fazer buscas na web
- Resultados devem ser mostrados por fala
- Pesquisa de imagem
- Pesquisa de video
- Pesquisa de filme
- Pesquisa de pdf
- Reconhecimento de multiplos comandos

3.1.2 Requisitos Não Funcionais

- Linguagem python V. 3.0
- Windows 10
- Dual Core 2.0 Ghz 6GB Ram
- Deve ser utilizado 2 nucleos
- A memoria nã o deve influenciar
- A conexão usada foi de 200 Mbs
- Não será necessario banco de dados, a não ser para armazenamento de log.
- A ideia inicial é que funcione para computadores pessoais, com potencial de escala para dispositivos moveis como tablets, celulares e iot.

3.2 Arquitetura

Varios modelos de arquiteturas de software se encaixam nesse projeto, então combinamos a ideia de camadas e microserviços para criar um tipo de arquitetura que tenha a modularidade e que possa ser montada em camadas assim o projeto pode ser levado para diferente plataformas.



Fonte: Autor, 2022.

A arquitetura foi divida em três partes, sistema de tempo real, classificador e execução de funções, cada uma tem um papel importante na busca das informações.

3.3 Sistema de reconhecimento Continuo

Um sistema de reconhecimento continuo tem o objetivo de encapsular a arquitetura basica e mante-lo se repetindo e sem que ocorram falhas que possam impedir o funcionamento do programa.

O processo se inicia executando um laço de repetição infinito, que fica executando a leitura do sensor (Microfone) pra saber quando deve guardar as informações lidas na variavel do texto de entrada.

```
#FUNÇÃO PRINCIPAL DE RECONHECIMENTO CONTINUO
def menu():
    i = " "
    #LAÇO DE REPETIÇÃO
    while i != "Sair":
        #LEITURA DO SENSOR MICROFONE
        resposta = Microfone()
        #DEFINIMOS UM NOME PARA NO ASSISTENTE,
        #COM PARAMETRO CONDICIONAL
        if resposta == "Nero":
            #CONTEUDO A SER RESPONDIDO
            #PELO SINTETIZADOR
            resposta = "Como posso ajudar?"
            #CHAMADA DO SINTETIZADOR
            sintetizador(resposta)
        else:
            #NÃO FARA NADA
        i = resposta
            #VARIAVEL DE CONTROLE DE SAIDA
```

Fonte: Autor, 2022

Essa função deve possuir um tratamento de exceções para que sejam evitados qualquer erro de software, que impessa seu bom funconamento. A primeira função a ser chamada sempre será a do MICROFONE, ela que decidira se tem ou não uma informação que deve ser chamada atravez de da palavra CHAVE.

3.3.1 Reconhecimento de Voz

O reconhecimento de voz é feito por uma biblioteca chamada SpeechRecognition. Ela foi escolhido para nossa implementação, por ser gratuita, possui alta precisão no reconhecimento de palavras e ainda possui atualizações.

```python
import speech_recognition as sr

#FUNÇÃO QUE RECONHECE O AUDIO DO MICROFONE
def Microfone():
    #FUNÇÃO QUE QUE INICIALIZA O MICROFONE
    recon = sr.Recognizer()
    #MELHOR METODO DE ABRIR E FECHAR  OS ARQUIVOS
    with sr.Microphone() as source:
        #RESPONSAVEL POR ELIMINAR RUIDOS
        recon.adjust_for_ambient_noise(source, duration
        recon.pause_threshold = 1
        #PODEMOS ESCOLHER LIMITE DE TEMPO QUE O MICROFO
        audio = recon.listen(source, timeout=7, phrase_
        #PODEMOS DEFINIR UM IDIOMA A SER IDENTIFICADO
        resposta = recon.recognize_google(audio, langua
        #TRATAMENTO DE EXCEÇÃO PARA QUE O MICROFONE NÃO
        try:
            resposta = recon.recognize_google(audio, la
            print(resposta)
        except WaitTimeoutError:
            print("SEM AUDIO - 1")
            Microfone()
            return "none"
        except Exception as e:
            return "none"
        return resposta
```

Fonte: Autor, 2022.

Essa função do microfone fica encapsulada e pode ser chamada em todas as outras funções, com necessidade apenas de receber o retorno da função, ou seja, receber o texto que foi dito no microfone e que agora fica armazenado na variavel.

Podemos também controlor o nivel de ruido, tempo de escuta e o idioma a ser reconhecido pelo programa. Todas essas tecnicas podem otimizar o desempenho fazendo o programa entender as sentenças com uma velocidade maior.

Trabalhar com reconhecimento de voz envolve varias problematicas então antecipamos os problemas, acompanhando o que ja é possivel identificar como falha, por exemplo tempo de leitura e erro de conexão, sendo essa biblioteca necessita de acesso de internet para fazer o reconhecimento na api do google, então conseguimos contornar o problema de tempo ignorando o seu retorno e quando não for possivel se conectar ele fara uma nova tentativa em recursividade.

A função de reconhecimento de voz é chamada dentro da função de reconhecimento continuo, então a cada loop a variavel captura o audio e faz uma comparação condicional para saber se deve ou não ir para o classificador.

Existem duas formas para validar a busca:

A - O nome da assistente deve ser chamada e então ela pergunta o que voce deseja fazer.

B – Chamando o nome da assistente com o argumento da função desejado.

Quando o condicional não reconhece o texto contido na variavel é ignorado, então o processo volta ao inicio do loop e a função do microfone se repete novamente até que o conteudo da varivel seja validado com a Chave.

A função quando for valida, retornara um mensagem de voz sintetizada avisando o que deve ser dito ou se vai ser executado algo, resuntante da busca.

3.3.2 Sintetizador de Voz

O sintetizador faz parte das funções extenas, apesar de estar dentro da função de reconhecimento continuo, quando uma função é executada ou ocorre algum erro o usuario é avisado atraves dessa função.

Para criar essa função usamos a biblioteca SSTX, ela possui uma base ja treinada que converte o texto dentro da variavel em voz, podendo configurar idioma, velocidade e a voz.

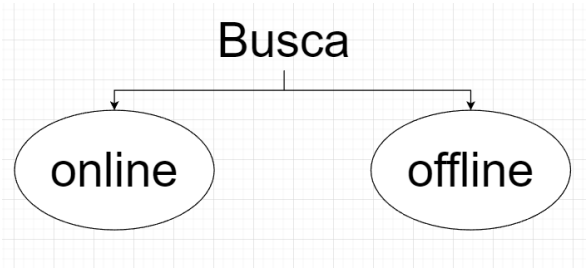
```python
import pyttsx3

#Função que faz a fala sintetica
def sintetizador(resposta):
    #inicializa os modulos
    en = pyttsx3.init()
    #escolha do idioma
    en.setProperty('voice', b'brazil')
    #velocidade de reprodução
    en.setProperty('rate', 190)
    #volume
    en.setProperty('volume', 1)
    #saida do audio
    en.say(resposta)
    #encerra a reprodução
    en.runAndWait()
```

Fonte: Autor, 2022.

3.3 Classisficador

O classificador possui duas ramificações online e offline. Essa separação foi feita para que a busca tenha um tempo de execução reduzido.



Fonte: Autor, 2022.

O classificador online é responsavel por executar funções que exigem buscas na internet, abrir navegador ou baixar alguna informação da rede.

O classificador offline gerencia as funções locais, como abrir uma pasta, copiar ou colar, abrir e fechar um programa.

Esse ideia de separação vem do conceito de arvores binarias, ou seja, a divide a informação para que consequentemente o caminho de busca da informação se pequeno, assim conseguimos encontrar a informação por um caminho menor.

Essa classificação segue uma sequências de procedimentos:

1. Dividir o texto em uma lista:

   Quando recebemos o pedido do usuario, toda a mensagem dita é guardada em uma string, e essa sentença deve ser tratada como uma lista de palavras, então podemos fazer as analises isoladamente.

2. Retirar a palavra chave do texto:

   Precisamos entender e pegar o conteudo principal da pesquisa, pois assim conseguir segmentar qual o tipo de pesquisa que o usuario está querendo. Então aplicamos o reconhecimento de palavra chave atraves de categorias previamente definadas na base de dados.

3. Fazer busca de sinonimos:

   As categorias que estão mapeadas nem sempre serão iguais as que o usuario vai dizer, então é importante identificar através de palavras semelhantes.

4. Identificar as funções:

   Depois que forem gerados os sinonimos as funções fazem comparações para identificar quais finções foram identificadas, para que o problema seja modularizado conforme o pedido do usuario.

5. Identificar os argumentos

Usando também a base de sinonimos podemos obter o conteudo atraves de eliminação de palavras. Isso é util pois as funções catalogadas serão encontradas e eliminadas na proucura dos argumentos.

6. Chamar o Núcleo de Execução

   Quando todos esse procedimento estiverem sido completados o nucle de funções deve, receber os nome funções e os argumentos das mesmasm, depois é aplicado uma ordenação de funções para que a busca seja a mais precisa possivel. Em seguida executamos todas as funções que foram montadas dentro da execução.

3.4 Nucleo de execução

O nucleo é composto por 3 funções, a primeira recebe os nomes das funções e os argumentos, a segunda ordena as funções e a terceira executa todas as funções localizadas na busca.

Quado chamada essa função recebemos a passagem de dois parametros, os nome da funções e os argumentos dela. O proximo passo é ordenar as funções.

A função chamadas estão armazenadas dentro de uma lista e cada uma tera uma função dentre pesquisar, buscar um video ou uma imagem e etc.

4 – Análise de resultados

Analisando os resultados, conseguimos implementar a pesquisa utilizando a base do wikipedia, fazer pesquisas diretas utilizando filtros atraves de webscrapping no google, podemos abrir imagens, videos e filmes fazendo uma pesquisa direta e clara.

O reconhecimento de voz por utilizar uma api online gera um pequenoo delay, mas não ao ponto de se tornar algo demorado.

As informações buscadas tem umm percentual de precisão, em quase todas as buscas conseguimos encontrar o conteudo e um resultado relevante.

As execuções das multiplas função está funcionando como funções compostas, ou seja, no momento ainda é necessario esperar a assistente pedir a proxima informação.

5 – Conclusôes

Ao longo desse trabalho criamos uma modelo de arquitetura, criado para ser implementado com futuras melhorias em assistentes virtuais, percebemos atraves dessas analises que a execução de multiplas funções é um ponto chave para que futuramente as asssistentes se tornem velozes e precisas na busca de informação, facilitando o acesso de dados a todas as pessoas, que tem necessidades especiais e para que seja facil para as outras também.

Conceguimos com esse estudo validar diversas ideias que ainda não estão sendo implementadas no mercado atual de assistentes.

Portanto conseguimos alcançar o nosso objetivo que é possivel implementar funções complexas para assistentes e com isso criar um novo modelo de busca automatizando processos e melhorando o acesso as pesquisas no geral.

REFERÊNCIAS

Olhar Digital. Notícias. Disponível em: <https://olhardigital.com.br/2020/10/24/noticias/como-surgiram-e-quais-sao-os-principais-assistentes-inteligentes/#:~:text=Saltando%20para%20a%20d%C3%A9cada%20de,para%20fazer%20pesquisas%20na%20web.>. Acesso em: Setembro de 2022.

Alexandra Lorandi Macedo e col. UM ESTUDO SOBRE A INTRODUÇÃO DE UM ASSISTENTE VIRTUAL PARA SUPORTE À ESCRITA COLETIVA. Rio grande do Sul , 2007.

Giovanna Dourado Marchi e col. Você Nunca Mais Estará Sozinho: uma análise dos relacionamentos com inteligências artificiais a partir do filme Her. Pernambuco, 2019.

Daiane Rose da Silva Matos, Francisco Kelsen de Oliveira. Análise com assistentes virtuais inteligentes: Um estudo decaso com o Google Assistente. Pernambuco, 2011.

Ingredy Anhaia e col. ASSISTENTE VIRTUAL "E AÍ LUNA". Fortaleza, 2018.

Leoncio Teixeira Cruz e col. Assistentes Virtuais Inteligentes. 1ª edição. Brasport Livros e Multimídia Ltda, 2013.

Jean-Marie Farines e col. Sistemas de tempo real. Florianópolis, julho de 2000.

Vladimir Fabregas Suringué de Alencar. Atributos e Domínios de Interpolação Eficientes em Reconhecimento de Voz Distribuído. Rio de Janeiro, março de 2005.

Silberschatz, Abraham, et al. Fundamentos de Sistemas Operacionais. Disponível em: Minha Biblioteca, (9th edição). Grupo GEN, 2015.

Fabíola Pantoja Oliveira Araújo. Imitação da Voz Humana através do Processo de Análise-por-Síntese utilizando Algoritmo Genético e Sintetizador de Voz por Formantes. Pará, 2015.