

Assistente Virtual por Comando de Voz para Apoiar o Ensino e a Autonomia de Estudantes com Deficiência Motora

Ana Mara de Oliveira Figueiredo, Francisco Freitas,
Gian Ximenes Verdan Pontes, Orlando Pereira Afonso Júnior

¹Instituto Federal Fluminense Campos Bom Jesus do Itabapoana

{ana.figueiredo, kiko ,ojunior}@iff.edu.br

gianverdan@gmail.com

Abstract. *Technological advances have expanded the use of computational systems, yet people with motor disabilities still face barriers in accessing these tools. This work proposes a voice-command virtual assistant, developed in Python, to promote digital inclusion. The solution uses speech recognition libraries and enables tasks such as opening applications, browsing the internet, and moving the cursor. The methodology involved exploratory research, iterative development, and practical testing. Results demonstrate the prototype's effectiveness in controlled environments. It is concluded that the tool is promising as a foundation for more accessible and customizable assistive technologies.*

Resumo. *O avanço tecnológico ampliou o uso de sistemas computacionais, mas pessoas com deficiência motora ainda enfrentam barreiras no acesso a essas ferramentas. Este trabalho propõe um assistente virtual por comando de voz, desenvolvido em Python, para promover inclusão digital. A solução utiliza bibliotecas de reconhecimento de fala e permite realizar tarefas como abrir aplicativos, navegar na internet e mover o cursor. A metodologia envolveu pesquisa exploratória, desenvolvimento iterativo e testes práticos. Os resultados demonstram a eficácia do protótipo em ambiente controlado. Conclui-se que a ferramenta é promissora como base para tecnologias assistivas mais acessíveis e personalizáveis.*

1. Introdução

A tecnologia digital está cada vez mais presente no cotidiano, impactando positivamente diversas áreas da sociedade. Contudo, ainda existem grupos que enfrentam barreiras para acessar essas tecnologias, como pessoas com deficiência motora. A exclusão digital desses indivíduos reforça desigualdades e limita oportunidades de inclusão social e profissional.

Dentre os recursos de acessibilidade existentes, os assistentes virtuais inteligentes (AVIs) se destacam como ferramentas promissoras, especialmente quando integrados a sistemas de comando por voz. No entanto, as soluções comerciais atuais geralmente têm suporte limitado a idiomas, funcionalidades restritas e pouca personalização.

No âmbito educacional, essas barreiras se manifestam de forma ainda mais crítica. Atividades rotineiras para um estudante, como a participação em Ambientes Virtuais de

Aprendizagem (AVAs), a elaboração de trabalhos acadêmicos ou mesmo a leitura de artigos em formato digital, exigem interações complexas e repetitivas com o teclado e o mouse. Para um estudante com limitações motoras, tais tarefas podem representar um obstáculo significativo, demandando grande esforço e limitando seu potencial de aprendizado e autonomia. Nesse cenário, um Assistente Virtual voltado ao ensino deve ir além de comandos genéricos, oferecendo funcionalidades específicas como: integração com plataformas educacionais para navegar por disciplinas e entregar tarefas, ferramentas de formatação de texto por voz e, principalmente, a capacidade de ler e interagir com materiais didáticos digitais de forma fluida.

Neste contexto, o presente artigo propõe o desenvolvimento de um assistente virtual por comando de voz com foco na inclusão de pessoas com deficiência motora. A proposta visa permitir a interação com sistemas operacionais de forma simplificada e adaptável às necessidades do usuário.

A metodologia adotada baseou-se em etapas iterativas de pesquisa, desenvolvimento e testes, utilizando a linguagem Python e bibliotecas de reconhecimento de voz. Foram implementadas funcionalidades como abertura de aplicativos, pesquisa na web e movimentação do cursor por voz.

Os resultados obtidos mostram que o protótipo é funcional e personalizável, podendo ser facilmente ampliado. Tais resultados apontam para o potencial da proposta como base para tecnologias assistivas acessíveis e abertas.

Este artigo apresenta como principais contribuições: (i) o desenvolvimento de uma solução acessível e de código aberto para inclusão digital; (ii) a demonstração prática da aplicação de técnicas de IA e reconhecimento de voz em contextos educacionais e sociais; (iii) o estímulo à adoção de abordagens inclusivas no ensino de informática e/ou computação.

O artigo está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados; a Seção 3 apresenta a metodologia utilizada; a Seção 4 descreve os resultados obtidos; e a Seção 5 traz as conclusões e perspectivas futuras.

2. Trabalhos Relacionados

Pessoas com deficiência (PcD), em especial aquelas com limitações motoras, enfrentam inúmeras barreiras ao interagir com tecnologias computacionais. A acessibilidade digital ainda é limitada por fatores como alto custo de dispositivos adaptados, escassez de soluções nacionais e pouca flexibilidade de personalização. De acordo com dados atuais [de Geografia e Estatística (IBGE) 2022], cerca de 6,5 milhões de brasileiros possuem deficiência motora que afeta os membros superiores, o que impacta diretamente sua capacidade de interagir com computadores e acessar a internet.

Tecnologias Assistivas (TAs) têm como objetivo ampliar as capacidades funcionais das PcDs, promovendo sua autonomia e inclusão digital e educacional. Contudo, como observado por [Muniz and Stringhini 2020], mesmo soluções já disponíveis como rastreamento ocular ou sensores adaptativos esbarram em barreiras de custo e compatibilidade física com o perfil dos usuários. Diante disso, esforços têm sido feitos para desenvolver dispositivos de baixo custo e adaptados às reais necessidades dos usuários, como o protótipo Mouse Button, que reduziu significativamente cliques acidentais e frustrações

em PcDs com espasmos musculares involuntários.

No campo educacional, projetos como o da APAE de Serra Talhada, [Muniz and Stringhini 2020], demonstram o impacto de ações integradas que envolvem ensino de informática, robótica e pensamento computacional para PcDs. Tais ações fortalecem o papel das TAs não apenas como ferramentas de suporte, mas como facilitadoras do processo de aprendizagem e inclusão sociotécnica.

Com o avanço da Inteligência Artificial, surgem novas possibilidades de interação adaptada com sistemas computacionais. Em especial, os Assistentes Virtuais Inteligentes (AVIs) mostram-se promissores para auxiliar PcDs, sobretudo quando integrados a comandos de voz. [Santana and Santos 2021] apontam que, para PcDs, os AVIs não são apenas facilitadores, mas verdadeiras pontes para independência, especialmente em contextos educacionais e de uso doméstico.

Além disso, o uso de Modelos de Linguagem de Grande Escala (LLMs) tem permitido melhorias significativas na personalização da interação homem-máquina. O trabalho de [Livero and Silva 2024] mostra que AVIs baseados em LLMs podem ser adaptados para diferentes públicos com deficiência, inclusive pessoas com deficiência visual, integrando entrada e saída por áudio e oferecendo suporte contextualizado e modular.

A base técnica do presente trabalho se ancora em recursos de reconhecimento de fala, viabilizados por técnicas de Aprendizado de Máquina (machine learning) e Aprendizado Profundo (deep learning). [Dörr and Aylon 2022] explicam como essas tecnologias permitem que os sistemas aprendam padrões a partir de dados sonoros e os interpretem como comandos úteis, expandindo o potencial de acessibilidade.

No cenário do desenvolvimento de software, linguagens de programação como Python oferecem bibliotecas acessíveis e de código aberto — como SpeechRecognition, Pytsx3, PyAudio — que facilitam a construção de ferramentas inclusivas. [Marques 2016] destaca como o uso dessas ferramentas abertas contribui para um ecossistema colaborativo e economicamente viável, fundamental no desenvolvimento de soluções assistivas em ambientes educacionais e de pesquisa.

Apesar do avanço de Assistentes Virtuais Inteligentes (AVIs) comerciais, como Google Assistant, Amazon Alexa e Siri, sua aplicação como tecnologia assistiva para controle computacional ainda é limitada. Tais sistemas são projetados para tarefas de propósito geral, como buscas de informação e controle de dispositivos domésticos, mas carecem de funcionalidades essenciais para a autonomia de usuários com deficiência motora severa, como o controle do cursor e a interação direta com elementos de software de terceiros. Além disso, por serem soluções de código fechado, oferecem pouca ou nenhuma margem para personalização de comandos específicos, adaptação a necessidades particulares ou integração com softwares educacionais, uma barreira significativa para o contexto proposto neste trabalho.

Assim, o presente trabalho se diferencia ao propor uma solução prática, de baixo custo, personalizável, com foco na inclusão digital de PcDs motoras, utilizando bibliotecas de código aberto, comandos por voz em português e estrutura modular para futura expansão.

3. Metodologia

A metodologia adotada neste trabalho foi dividida em duas etapas principais: teórica e prática, seguindo um fluxo iterativo de pesquisa, desenvolvimento e testes, como mostrado na 1.

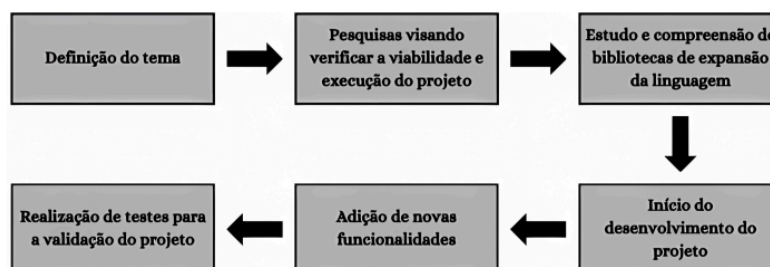


Figura 1. Esquema metodológico adotado neste trabalho

3.1. Primeira Etapa - Teórica

A primeira fase consistiu na definição do escopo do projeto, com a delimitação do problema enfrentado por pessoas com deficiência motora no uso do computador e a formulação dos objetivos do assistente virtual por comando de voz.

Foi realizada uma pesquisa exploratória, com levantamento teórico sobre tecnologias assistivas, aplicações de inteligência artificial na acessibilidade e fundamentos sobre reconhecimento de voz. Em paralelo, foram analisadas as possibilidades de implementação com base em viabilidade técnica e acessibilidade dos recursos.

Na sequência, foi feito um estudo aprofundado das bibliotecas utilizadas para implementação em Python, considerando aspectos como compatibilidade com o sistema operacional, licenças de uso e funcionalidades disponíveis. Dentre as bibliotecas estudadas destacam-se: SpeechRecognition, Pytsx3, PyAudio, entre outras.

3.2. Segunda Etapa - Prática

Com a viabilidade confirmada, iniciou-se a fase de desenvolvimento. O primeiro passo foi a implementação do reconhecimento de voz por meio do microfone. Com essa funcionalidade estabilizada, foram adicionadas as funções básicas do assistente virtual, como informar a hora, tocar músicas, abrir programas e realizar pesquisas.

Para permitir a execução contínua e simultânea de múltiplas funcionalidades, o sistema foi estruturado com um loop de reinicialização automática e com ativação na inicialização do sistema operacional.

As funcionalidades são ativadas por palavras-chave específicas, conforme apresentado na Tabela 1. Comandos de teclado foram integrados para permitir maior controle do sistema e, para garantir adaptabilidade, foi implementado um arquivo de configuração, no qual o usuário pode ajustar os atalhos e remover artigos desnecessários de pesquisas por voz.

Testes práticos foram conduzidos para validar a precisão do reconhecimento de comandos e a efetividade da execução de tarefas, em diferentes contextos de uso.

Tabela 1. Funções do assistente virtual e palavras de ativação

Função	Descrição	Palavra de ativação
Horário	Informa ao usuário o horário atual.	'horas'
Pesquisa com resposta por voz	Busca na <i>Wikipédia</i> mais informações sobre algo de interesse do usuário.	'procure por'
Reprodutor de música	Reproduz uma música pedida pelo usuário.	'toque'
Desligar o computador	Desliga o computador por meio dos comandos por voz.	'desligar PC'
Abertura de aplicativos	Abre aplicativos como o Google Chrome.	nome do aplicativo
Utilização do navegador	Permite a utilização do navegador por comando de voz possibilitando pesquisar, navegar por sites e guias etc.	'pesquise', 'nova guia', 'próxima guia', etc.
Utilização do mouse	Faz a utilização das funções do mouse como a movimentação ou clique.	'mouse' + direção e sentido do movimento
Criação de lembretes	Faz o uso da agenda do google para a criação de lembretes.	'agenda'
Previsão do tempo	Fornece a previsão do tempo de acordo com a cidade fornecida com o usuário	'tempo'

4. Resultados e Discussões

O assistente virtual desenvolvido demonstrou desempenho satisfatório na execução de comandos por voz, respondendo de forma ágil e precisa em ambiente controlado. Esta seção apresenta as principais funcionalidades implementadas, acompanhadas de trechos de código e imagens que ilustram seu funcionamento.

4.1. Abertura de Aplicativos por Comando de Voz

O sistema é capaz de reconhecer o nome de um aplicativo mencionado pelo usuário e realizar sua abertura automaticamente. Por exemplo, ao dizer “Google”, o assistente identifica o comando e inicia o navegador Google Chrome (desde que esteja instalado no sistema).

```
elif "google" in comando:
    print("Abrindo o google")
    os.system('"C:\Program Files\Google\Chrome\Application\chrome.exe"')
    maquina.runAndWait()
```

Figura 2. Trecho de código responsável pela abertura de aplicativos por nome

Essa funcionalidade favorece a autonomia de usuários com deficiência motora, permitindo o acesso a softwares sem a necessidade de utilizar mouse ou teclado.

4.2. Pesquisa Automatizada com Remoção de Artigos

Ao detectar um comando iniciado pela palavra “pesquisa”, o assistente realiza uma limpeza textual, removendo artigos e conectores que possam comprometer a clareza da busca. Em seguida, executa a pesquisa no navegador utilizando os termos mais relevantes.

```

elif "pesquise" in comando:
    print("Pesquisando")
    pyautogui.hotkey(comandos["barra de pesquisa"][0], comandos["barra de pesquisa"][1])
    pesquisa = comando.replace("pesquise", "")
    pesquisa = pesquisa.replace("site", "")
    pesquisa = pesquisa.replace("abra", "")
    pesquisa = pesquisa.replace("entre", "")
    for a in artigos:
        pesquisa = pesquisa.replace(a, "")
    pyautogui.write(pesquisa)
    pyautogui.press("enter")
    maquina.runAndWait()

```

Figura 3. Trecho de código responsável pela realização de pesquisas com limpeza da linguagem

```

artigos = [
    "a ", " as ", " o ", " os ", " um ", " uma ", " uns ", " umas ", " no ", " na ", " nos ",
    " nas ", " do ", " da ", " das ", " dos ",
]

```

Figura 4. Lista de artigos desnecessários removidos das pesquisas por voz

Essa abordagem contribui para uma experiência de busca mais objetiva e eficiente, especialmente para usuários que utilizam comandos de voz com maior informalidade.

4.3. Abertura de Nova Guia no Navegador

O comando “nova guia” aciona a simulação de atalhos de teclado, permitindo a abertura de uma nova aba no navegador sem intervenção manual.

```

elif "nova guia" in comando:
    #pyautogui.hotkey("ctrl", "t")
    print("Abrindo uma nova guia")
    pyautogui.hotkey(comandos["nova guia"][0], comandos["nova guia"][1])
    maquina.runAndWait()

```

Figura 5. Trecho de código para simulação de atalho de abertura de nova guia

Esse recurso amplia as possibilidades de navegação para o usuário, promovendo continuidade e fluidez durante o uso da internet.

4.4. Movimentação do Cursor por Voz – Direita

O assistente interpreta comandos como “mouse direita três”, nos quais a direção e a intensidade são traduzidas em pixels para movimentação do cursor.

Essa funcionalidade é essencial para permitir a navegação em sistemas operacionais e páginas web sem o uso do periférico físico.

4.5. Movimentação do Cursor – Esquerda com Intensidade Aumentada

Com comandos similares ao anterior, o usuário também pode realizar movimentações mais longas do cursor. Por exemplo, “mouse esquerda cinco” desloca o ponteiro com maior intensidade.

Essa flexibilidade na movimentação permite a adaptação do assistente a diferentes resoluções de tela e níveis de mobilidade do usuário.

```

elif "mouse" in comando:
    #comando para mover muito, um pouco ou regularmente o mouse para a direita
    if "direita" in comando:
        if "um pouco" in comando:
            print("Movendo um pouco o mouse para a direita")
            pyautogui.moveRel(largura * 0.04, 0)
            maquina.runAndWait()
        elif "muito" in comando:
            print("Movendo muito o mouse para a direita")
            pyautogui.moveRel(largura * 0.5, 0)
            maquina.runAndWait()
        else:
            print("Movendo o mouse para a direita")
            pyautogui.moveRel(largura * 0.3, 0)
            maquina.runAndWait()

```

Figura 6. Trecho de código para movimentação do mouse para a direita

```

elif "esquerda" in comando:
    if "um pouco" in comando:
        print("Movendo um pouco o mouse para a esquerda")
        pyautogui.moveRel(largura * -0.04, 0)
        maquina.runAndWait()
    elif "muito" in comando:
        print("Movendo muito o mouse para a esquerda")
        pyautogui.moveRel(largura * -0.5, 0)
        maquina.runAndWait()
    else:
        print("Movendo o mouse para a esquerda")
        pyautogui.moveRel(largura * -0.3, 0)
        maquina.runAndWait()

```

Figura 7. Trecho de código para movimentação do mouse para a esquerda com maior intensidade

4.6. Adaptação e Configurabilidade

O sistema inclui um arquivo externo de configuração que centraliza os parâmetros de uso. Nesse arquivo, o usuário pode alterar palavras-chave, modificar atalhos e personalizar filtros de linguagem para as pesquisas por voz. É importante notar que, na versão atual do protótipo, essa personalização é realizada através da edição direta de um arquivo de configuração, não contando ainda com uma interface gráfica dedicada.

Essa característica torna o sistema altamente adaptável, possibilitando sua adequação a diferentes perfis e preferências. Além disso, a modularidade da implementação facilita a manutenção e evolução do projeto.

4.7. Etapas Futuras do Trabalho

Como desdobramento deste trabalho, planeja-se a validação empírica por meio de testes com um grupo de usuários reais que possuam diferentes graus de deficiência motora. O objetivo é avaliar a usabilidade do assistente em situações práticas, simulando tarefas edu-

cacionais como a navegação em ambientes de aprendizagem e a edição de textos. Para isso, será aplicado um protocolo de avaliação com métricas quantitativas (como a taxa de sucesso no reconhecimento de comandos, tempo médio de resposta e número de falhas) e qualitativas, baseadas em questionários de satisfação e na coleta de feedback espontâneo para identificar dificuldades e sugestões de melhoria. Essa etapa é fundamental para orientar o ciclo de desenvolvimento com base em evidências, garantindo que a evolução do projeto seja diretamente informada pelas necessidades de seus usuários. Os próximos passos para este plano incluem a definição de um cronograma detalhado e a busca por parcerias com instituições de apoio, como a APAE, para facilitar o recrutamento de participantes e a validação em um contexto real.

Por ser um projeto de código aberto, a solução proposta possui grande potencial de expansão. Planeja-se a integração direta com ambientes educacionais, desenvolvendo módulos capazes de interagir com plataformas educacionais como o Moodle ou Google Classroom e executar comandos específicos como abrir a página de uma matéria específica. Além disso o assistente pode ser adaptado para outras plataformas, integrar novos serviços (como previsão do tempo, controle de dispositivos IoT, leitura de e-mails, entre outros).

Do ponto de vista técnico e de acessibilidade, futuras investigações incluirão a análise comparativa de desempenho entre a biblioteca SpeechRecognition e alternativas mais recentes como a OpenAI Whisper. Serão exploradas também estratégias para usuários com dificuldades de fala, como o treinamento de modelos de reconhecimento personalizados, e a garantia de compatibilidade com leitores de tela para atender usuários que possuam deficiência visual associada.

Adicionalmente, um passo crucial para a evolução do projeto é o desenvolvimento de uma interface gráfica amigável para configuração. Visando aprimorar a usabilidade e a autonomia do usuário, essa interface substituirá a necessidade de edição manual do arquivo de configuração, uma tarefa que pode apresentar barreiras para pessoas sem conhecimento técnico. Por meio dela, o usuário poderá ajustar de forma intuitiva parâmetros essenciais como a sensibilidade e a velocidade de movimentação do cursor, personalizar a lista de artigos a serem removidos das pesquisas por voz e remapear as palavras-chave de ativação para as diferentes funcionalidades do sistema. A implementação deste recurso é fundamental para consolidar a ferramenta como uma solução verdadeiramente acessível e adaptada às preferências de cada indivíduo.

O repositório do projeto será disponibilizado publicamente na versão final do artigo, respeitando os critérios de anonimização exigidos para o processo de revisão às cegas.

5. Conclusão

Este trabalho apresentou o desenvolvimento de um assistente virtual acessível por comando de voz, com o objetivo de promover a inclusão digital de pessoas com deficiência motora. Utilizando bibliotecas de código aberto em Python e técnicas de reconhecimento de fala, o sistema mostrou-se eficaz na execução de tarefas básicas do sistema operacional.

A proposta é relevante por sua aplicabilidade prática, simplicidade de implementação e potencial de personalização. Além disso, incentiva a reflexão sobre

o papel das tecnologias assistivas na redução das barreiras enfrentadas por pessoas com deficiência no acesso à informação e ao uso de computadores.

As perspectivas futuras, já detalhadas na seção anterior, envolvem o aprimoramento iterativo da ferramenta com base na validação empírica junto ao público-alvo e na expansão de suas funcionalidades, consolidando-a como uma plataforma assistiva de código aberto, adaptável e de grande impacto social.

Referências

- de Geografia e Estatística (IBGE), I. B. (2022). *Censo Demográfico 2022*. IBGE. Consultado em 09 de Junho de 2025.
- Dörr, J. B. and Aylon, L. B. R. (2022). Um estudo sobre técnicas utilizadas para o reconhecimento de sons com o uso de inteligência artificial e python. In *Anais do XIX Congresso Latino-Americano de Software Livre e Tecnologias Abertas*, pages 103–112. SBC.
- Livero, N. J. and Silva, F. S. (2024). Desenvolvimento de um assistente virtual baseado em voz e llms para facilitar a interação de estudantes com deficiência visual com sistemas operacionais. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 2987–2998. SBC.
- Marques, R. (2016). Api, bibliotecas e frameworks: entenda a diferença entre eles. <https://www.cedrotech.com/blog/api-bibliotecas-e-frameworks-entenda-diferenca-entre-eles/>. Acesso em: 20 set. 2023.
- Muniz, I. R. and Stringhini, D. (2020). Desenvolvimento de protótipos de hardware com baixo custo para inclusão digital de pessoas com deficiência motora. In *Simpósio Brasileiro de Computação Aplicada à Saúde (SBCAS)*, pages 105–110. SBC.
- Santana, B. B. and Santos, M. C. D. A. (2021). Assistente virtual inteligente e suas aplicações na sociedade. *Coruja Informa*. <http://www.each.usp.br/petsi/jornal/?p=2813>. Acesso em: 20 set. 2023.