

Universidade Cruzeiro do Sul

Henrik Beck

Victor Henrique Ranalli Barbosa

Construção de uma assistente virtual para controlar o gerenciador de janelas i3WM no
Linux através de comandos por voz

São Paulo

2022

Henrik Beck
Victor Henrique Ranalli Barbosa

Construção de uma assistente virtual para controlar o gerenciador de janelas i3WM no
Linux através de comandos por voz

Trabalho de graduação
interdisciplinar (parte 2) destinado a
Universidade Cruzeiro do Sul como parte
das exigências para a obtenção do título de
bacharel em Ciências da Computação.

Orientador Prof^o. Manuel F.
Paradela Ledón

São Paulo
2022

RESUMO

O desenvolvimento e a configuração de uma assistente virtual é um meio de proporcionar uma ferramenta de integração entre o sistema operacional e humanos, um cenário na qual pessoas que possuem mobilidade reduzida possam se beneficiar. Disponibilizar um leque maior de possibilidades a respeito de ferramentas de acessibilidade de software pode fazer com que o usuário decida qual delas seja melhor para sua individualidade. A existência dos recursos de controle do comportamento de janelas, abertura e fechamento de programas, além do gerenciamento de controle multimídia por meio da interação por voz são alguns dos benefícios a serem desfrutados deste software. O objetivo deste projeto consiste na construção de uma assistente virtual para controlar o gerenciador de janelas i3WM no Linux através de comandos por voz. Para esta tarefa está sendo utilizado a linguagem de programação *Python* em conjunto com bibliotecas de código fonte aberto que sejam capazes de usufruir do sistema de controle de áudio do computador; reconhecer e transcrever a voz humana em texto; e, fazer o computador pronuncie palavras a partir de textos.

Palavras chaves: Assistente virtual, i3wm, Linux, Python.

ABSTRACT

The development and configuration of a virtual assistant is a way to provide an integration tool between the operating system and humans, a scenario whose people with reduced mobility can benefit. Providing more possibilities regarding software accessibility tool can make the user decide which one is best for his individuality. The existence of features for controlling the behavior of windows, opening and closing programs, in addition to managing multimedia control through voice interaction are some of the benefits to be enjoyed from this software. The objective of this project is to build a virtual assistant to control the i3WM window manager on Linux through voice commands. For this task, the Python programming language is being used in favor of open source libraries whose are able to take some advantage of the computers audio control system; recognize and transcribe the human voice into texto; and make the computer pronounce words from texts.

Keywords: Virtual assistant, i3wm, Linux, Python.

SUMÁRIO

1 INTRODUÇÃO.....	5
1.1 CAPTURA DE ÁUDIO.....	5
1.2 ACESSIBILIDADE.....	5
1.3 LICENÇA DE USO.....	5
1.4 LINUX.....	6
1.5 INTERFACE GRÁFICA.....	6
1.5.1 Conceito.....	6
1.5.2 Gerenciador de janelas i3wm.....	6
1.6 BIBLIOTECAS DE RECONHECIMENTO DE ÁUDIO.....	7
1.6.1 SpeechRecognition.....	7
1.6.2 Sphinx4.....	7
1.6.3 Watson Speech to Text.....	7
1.6.4 Critérios de escolha da biblioteca de reconhecimento de áudio.....	7
1.7 SINTETIZADOR DE VOZ.....	8
1.7.1 Pyttsx3.....	8
1.7.2 Festival Speech Synthesis System.....	8
1.7.3 Critérios de escolha do sintetizador de voz.....	8
1.8 INTEGRAÇÃO ENTRE A ASSISTENTE VIRTUAL E O SISTEMA OPERACIONAL.....	8
1.8.1 Shell Script Library.....	8
2 DESENVOLVIMENTO DO PROJETO.....	9
2.1 FUNCIONALIDADES DO PROJETO.....	9
2.2 ENGENHARIA DE SOFTWARE.....	9
2.2.1 Histórias de usuários.....	9
2.2.2 Regras de negócio.....	9
2.2.3 Requisitos funcionais.....	9
2.2.4 Requisitos não funcionais.....	9
2.2.5 Diagrama de caso de uso.....	10
2.2.6 Diagrama de classe.....	11
2.2.7 Diagrama de componentes.....	12
2.3 DIFICULDADES ENFRENTADAS.....	13

2.3.1	Versionamento do Python.....	13
2.3.2	Ambientes virtuais.....	13
2.3.3	A solução.....	13
3	RESULTADOS.....	13
3.1	PRECISÃO DO RECONHECIMENTO DE VOZ.....	13
3.2	EXECUÇÃO DA ASSISTENTE VIRTUAL.....	13
3.3	LISTA DE COMANDOS SUPORTADOS.....	13
3.3.1	Ação de abertura de softwares.....	13
3.3.2	Ação do sistema de áudio.....	14
3.3.3	Ação do sistema de brilho da tela.....	14
3.3.4	Ação de controle da janela.....	14
4	DISCUSSÃO ACADÊMICA.....	15
5	CONCLUSÃO.....	15
6	APÊNDICE.....	16
6.1	PRIORIDADES.....	16
6.2	CATEGORIAS.....	16
6.3	HISTÓRIAS DE USUÁRIOS.....	17
6.4	REGRAS DE NEGÓCIO.....	18
6.5	REQUISITOS FUNCIONAIS.....	19
6.6	REQUISITOS NÃO FUNCIONAIS.....	19
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	20

1 INTRODUÇÃO

1.1 CAPTURA DE ÁUDIO

De acordo com Cardoso (2001), O reconhecimento de voz pode ser definido como a capacidade de um computador converter a sonoridade das palavras em código binário, o qual é compreensível pelo computador.

Partindo deste conceito, Alexandra et al. (2003) dizem que o som pode ser capturado a partir de um microfone simples no caso de um solução que não exige muita qualidade, ou através de uma mesa de som e amplificador para ambientes mais complexos possibilitando o balanceamento de sinal e ajuste de volume. Para a construção deste projeto nos concentramos em dispositivos de capturas de áudio simples, uma vez que tais recursos sejam mais acessíveis ao usuário comum perante ao custo de aquisição e o nível de complexidade de configuração.

Ao capturar o sinal de áudio para o computador, sua recepção digital frutifica um amplo panorama de possibilidades proporcionando inclusive a excentricidade de "conversar com a máquina". O valor de controlar o computador a partir de comandos de voz é deixar as mãos livres para outras funções, além de ser muito útil para propiciar maior acessibilidade às pessoas com necessidades especiais, pois oferece a oportunidade de usar o computador de uma forma que antes não podiam (SCOPE, s.d, apud DOS SANTOS, BERNAL, FERNANDEZ, 2018).

1.2 ACESSIBILIDADE

Em um levantamento realizado pelo Censo Demográfico de 2010 (IBGE, 2016), aproximadamente 23,9% da população se declaram com algum tipo de deficiência, seja visual, mental, motora ou auditiva, desta forma representando cerca de 46 milhões de brasileiros. Esta parcela populacional é diariamente afetada pelas barreiras de acessibilidade que permeiam o ambiente físico urbanista que vivemos e muitas vezes este cenário é estendido ao ambiente virtual por conta das limitações e, em particular a deficiência motora, cuja representatividade é de 6,95%, ou seja, aproximadamente 13 milhões de pessoas.

O Instituto de Tecnologia Social ITS (2008) apresentou a importância do desenvolvimento de ferramentas computacionais voltadas as necessidades das pessoas com deficiência. Uma vez que estas pessoas adquirem a independência da realização de tarefas computacionais básicas, como por exemplo, redigir textos e utilizar as redes sociais, elas sentem-se iguais aos demais usuários, aumentando assim sua autoestima através desta integração tecnológica.

1.3 LICENÇA DE USO

São muitas as maneiras pela qual um software pode ser licenciado, podendo ser essencialmente dividido entre os grupos de softwares proprietários e os de softwares livres. E esta questão das licenças de uso é muito importante para o desenvolvimento e a adoção de programas, em caráter especial, o software livre. Tais programas em geral são de simples obtenção, porém isto não significa que uma pessoa possa manipulá-lo da forma como queira (SABINO e KON, 2009).

Para Sabino e Kon (2009), a definição de código aberto não significa apenas o acesso ao código fonte havendo assim um conjunto de critérios conforme sintetizado a seguir:

1. Redistribuição livre.
2. Inclusão do código fonte.
3. Permitir modificações e trabalhos derivados.
4. Integridade do código fonte do autor.
5. Não haver discriminação às pessoas ou grupos.
6. Não haver discriminação a respeito das áreas comercialmente exploradas.

7. Os direitos associados ao programa devem ser aplicáveis a todos para quem o programa é redistribuído, sem a necessidade de execução de licenças adicionais para essas partes.
8. A licença de uso não deve ser específica a um produto, desta forma aplicado a mesma licença para suas partes e redistribuições.
9. A licença não deve restringir outro software.
10. A licença deve ser neutra as tecnologias ou estilo de interface.

A licença GNU (acrônimo de *GNU is Not Unix* - GNU não é Unix, em tradução livre) foi criada por Richard Stallman em 1983 e mantido atualmente pela *Free Software Foundation* (FSF) e concede quatro liberdades aos seus usuários, sendo elas: liberdade para executar o programa como quiser; liberdade para realizar quantas cópias julgar-se necessário; liberdade de realizar modificações no programa a próprio gosto; e a liberdade para distribuir versões melhoradas (SABINO, 2011). Para Gouvêa e Jamil, 2002, Richard Stallman tomou esta iniciativa com o intuito de que programas desenvolvidos por comunidades de programadores ou universidades fossem impedidos de seres capturados por grandes empresas da informática.

1.4 LINUX

Conforme os autores Ball e Buff (2004), Linux é o núcleo (ou *kernel* em inglês) de um sistema operacional gratuito, inspirado no Mini e que foi desenvolvido por Linus Benedict Torvalds no ano de 1991. Este é um programa executado no momento de inicialização do computador fornecendo uma interface entre o que é exibido ao usuário e o hardware. Este projeto está sob licença GNU, sendo assim muitas vezes chamado de GNU/Linux. Há diversos benefícios em utilizar o Linux, tais como o ótimo custo benefício, pois pode ter pouco ou nenhum custo por máquina e ser isento de taxas de royalties ou licenciamento. É um excelente uso tanto ambientes *desktop* quanto servidores.

Alguns dos exemplos de distribuições Linux existentes segundo Campos (2006), são o *Debian*, o *Fedora*, o *Gentoo*, o *Mandriva*, o *Red Hat Enterprise Linux*, o *Slackware* o *Ubuntu*, dentre outros. Contudo a distribuição *ArchLinux*, sob a óptica de Castro (2016), tem por objetivo alcançar os usuários avançado e tem um sistema de atualização nos molde de *rolling release* e seu sucesso está atrelado a sua simplicidade, pois trás por padrão um alicerce básico para que o usuário construa o sistema do jeito que preferir.

1.5 INTERFACE GRÁFICA

1.5.1 Conceito

“O computador e sua interface representam uma ferramenta cognitiva, uma extensão da memória, uma prótese cognitiva que permite tratar melhor a informação. É importante que se conheça como os processos cognitivos humanos se desenvolvem para a concepção de próteses cognitivas compatíveis com eles.”
(Cybis, 2003 P.3).

1.5.2 Gerenciador de janelas i3wm

O i3WM é um gerenciador de janelas que estende as funcionalidades do WMII e automaticamente posiciona as janelas organizando-as em um layout de árvore, na qual criam-se dois

frames e divide-os horizontalmente e verticalmente. Cada *frame* contém uma janela ou um outro par de *frames*. O layout de árvore pode ser redimensionado de acordo com as preferências do usuário, alterando assim o tamanho das janelas e cada área de trabalho virtual contém seus próprios *frames* e janelas (BRADLEY, 2018).

O mecanismo do i3 (o i3WM) pode ser controlado através de um *socket* UNIX e realiza a comunicação pelo utilitário *i3-msg*. Esta interface pode enviar qualquer comando disponível no arquivo de configuração do i3 para utilizar teclas de atalho (STAPELBERG, 2012).

1.6 BIBLIOTECAS DE RECONHECIMENTO DE ÁUDIO

1.6.1 SpeechRecognition

Uma biblioteca capaz de reconhecer discurso, com suporte a *APIs* online e offline para a linguagem *Python*. É uma biblioteca que está sob licença BSD e tem seu código fonte aberto (SpeechRecognition, 2019).

1.6.2 Sphinx4

Escrita completamente utilizando a linguagem de programação Java, o Sphinx-4 pode ser executado em uma variedade de plataformas sem requerir nenhuma compilação especial. É uma biblioteca que está sob licença BSD e tem seu código fonte aberto (SPHINX4, 2014).

1.6.3 Watson Speech to Text

Com o Watson Speech to Text (STT) é possível converter áudio em texto em mais de 15 idiomas incluindo Português, Inglês, Espanhol e Mandarim (IBM, 2021).

1.6.4 Critérios de escolha da biblioteca de reconhecimento de áudio

A Tabela 01 compara as biblioteca de reconhecimento de áudio.

Biblioteca	Suporte a idiomas inglês e português	Reconhecimento <i>online</i>	Reconhecimento <i>offline</i>	API gratuita
SpeechRecognition	Sim	Sim	Sim	Sim
Sphinx-4	Sim	Sim	Sim	Sim
Watson Speech to Text	Sim	Sim	Não	Até 500 requisições por dia

Tabela 01

Diante de tal comparação, optamos em utilizar a biblioteca SpeechRecognition para construção deste projeto.

1.7 SINTETIZADOR DE VOZ

1.7.1 Pyttsx3

Biblioteca que fornece uma coleção de drivers para Python para leitura de texto usando o sistema de áudio do sistema operacional (PYTTX3, 2017).

A Tabela 02 compara a relação entre o sistema operacional e o sintetizador de voz padrão (Adaptado de PYTTX3, 2017).

Sintetizador de voz	Sistema operacional
SAPI5	Windows XP, Windows Vista, Windows 8, 8.1 e 10
NSSpeechSynthesizer	Mac OS X 10.5 (Leopard) e 10.6 (Snow Leopard)
eSpeak	Ubuntu Desktop Edition 8.10, 9.04 e 9.10

Tabela 02

1.7.2 Festival Speech Synthesis System

Desenvolvido pelo departamento *The Centre for Speech Technology Research* (CSTR) da Universidade de Edinburgh. Este software foi escrito em C++ e está sob licença X11-type. Trata-se de um sistema de multilíngue que atualmente suporta inglês britânico, inglês americano, italiano, tcheco e espanhol (CSTR, 2008).

1.7.3 Critérios de escolha do sintetizador de voz

Primeiramente foi implementado a Pyttsx3 como sintetizador de voz padrão da assistente virtual. Porém, por motivos de incompatibilidade com versões superiores a 3.6 da linguagem Python, foi necessário efetuar sua troca para o software Festival.

1.8 INTEGRAÇÃO ENTRE A ASSISTENTE VIRTUAL E O SISTEMA OPERACIONAL

1.8.1 Shell Script Library

Trata-se de uma biblioteca escrita em Shell Script sob licença BSD que tem o intuito de abstrair comandos e funções do terminal do sistema operacional (BECK, 2022).

2 DESENVOLVIMENTO DO PROJETO

2.1 FUNCIONALIDADES DO PROJETO

O projeto desenvolvido por este trabalho tem como principal funcionalidade o comando de voz para controlar o gerenciador de janelas I3WM da interface gráfica I3. Sendo o sistema será capaz de:

1. Gravar a voz do usuário.
2. Pronunciar palavras/frases em inglês.
3. Reconhecer a voz do usuário em ambiente de conexão offline.
4. Reconhecer a voz do usuário no idioma inglês.
5. Reconhecer a voz do usuário no idioma português.
6. Controlar as janelas gráficas do gerenciador de janelas I3WM.
7. Abrir softwares instalados no sistema operacional.
8. Fechar softwares instalados no sistema operacional.

2.2 ENGENHARIA DE SOFTWARE

2.2.1 Histórias de usuários

Ver apêndices 6.3

2.2.2 Regras de negócio

Ver apêndice 6.4

2.2.3 Requisitos funcionais

Ver apêndice 6.5

2.2.4 Requisitos não funcionais

Ver apêndice 6.6

2.2.5 Diagrama de caso de uso

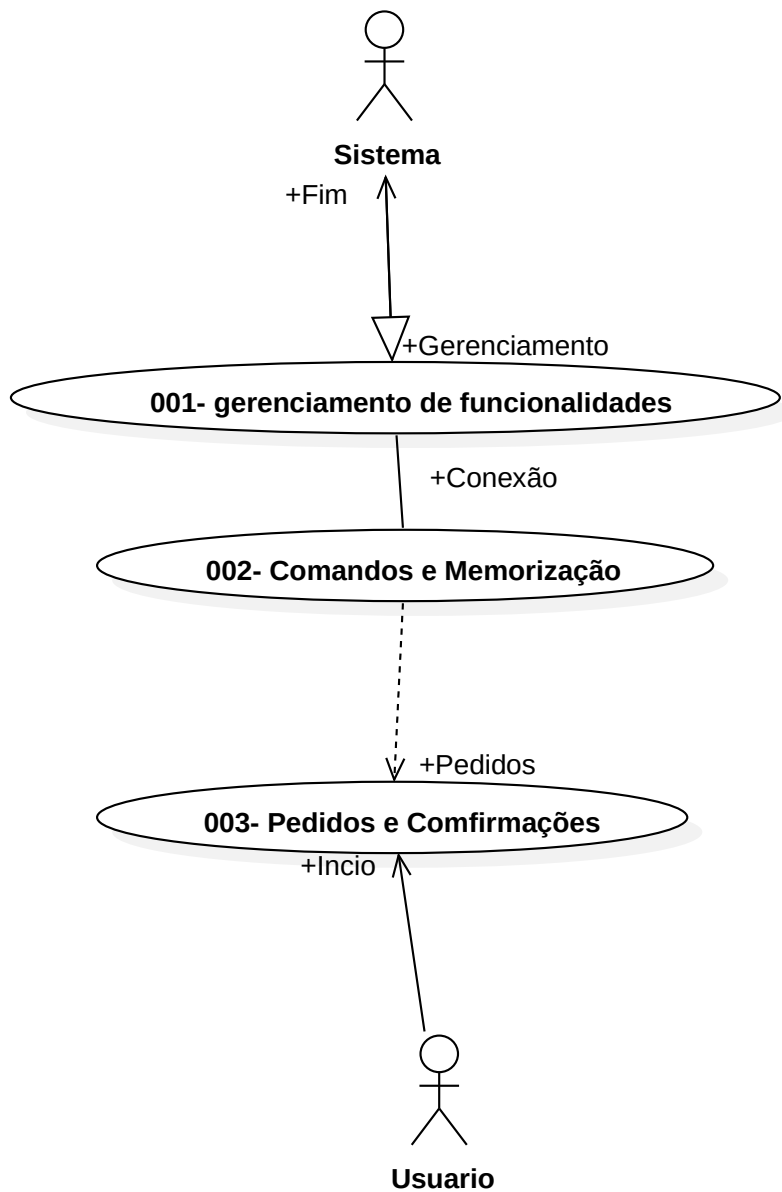


Figura 1

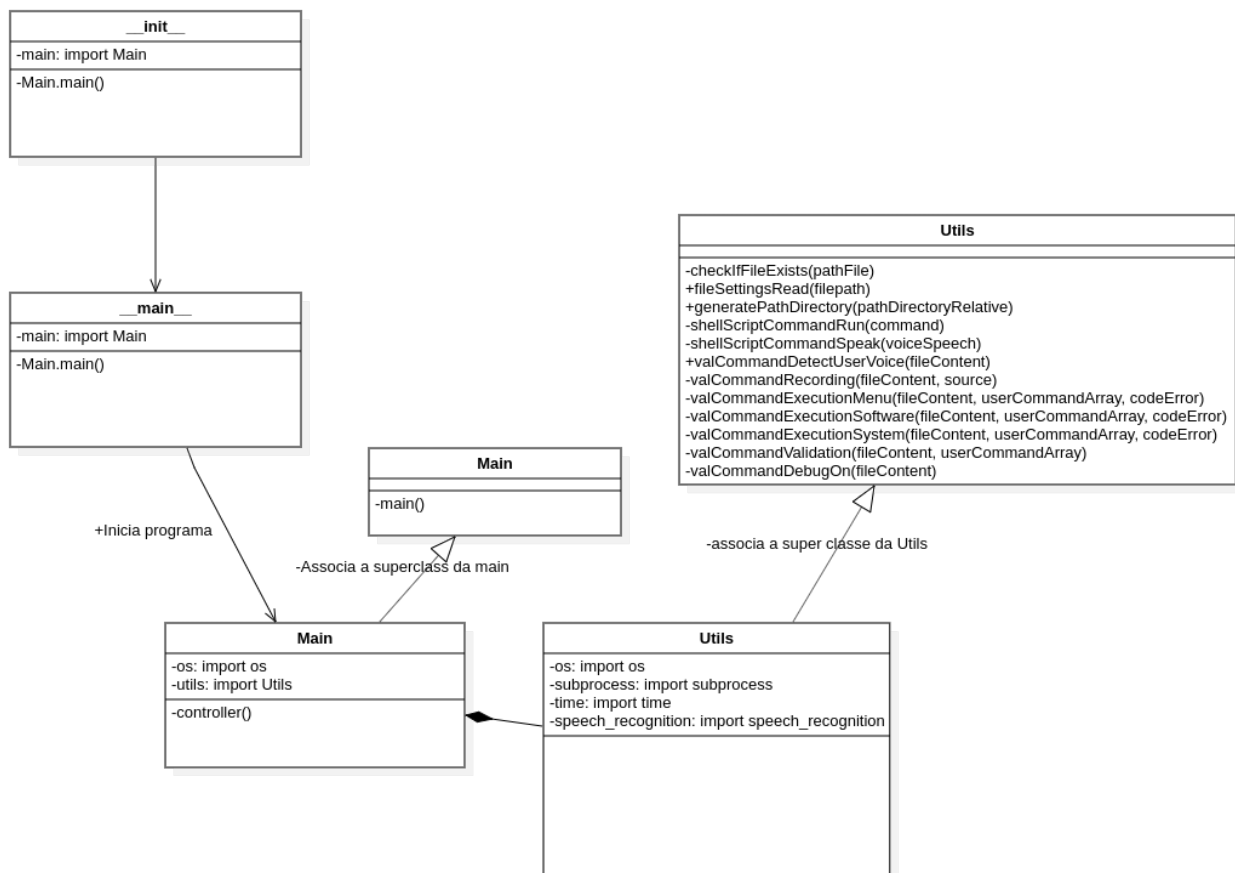
- **001 – Gerenciamento de funcionalidades**
Trata-se do caso de uso que verifica o pedido de seu cliente e aciona o sistema assim que verificado a sua integridade. Este caso de uso analisa o comando executado pelo usuário e em virtude da não existência do comando solicitado na memória, ele retorna falso.
- **002 – Comandos e memorização**
Etapa na qual prioriza-se a agilidade dentro do sistema onde é verificado se o usuário já solicitou ou não esta função. Se ele já pediu, não precisaremos pedir ao sistema

sua checagem esse comando já está dentro das normas. Ou seja ele será executado de uma forma ou de outra.

- 003 – Pedidos e confirmações

Aqui temos o clássico caso de uso da confirmação ou da **NEGAÇÃO**. Assim este caso é refere-se a realização caso o usuário solicitou um comando que a assistente virtual possa atender ou não. Desta forma retornam-se as respostas como dativas a ele, como por exemplo: “Este comando é inválido. Tente novamente”. Ou, em caso afirmativa, retorna-se a execução do comando solicitado.

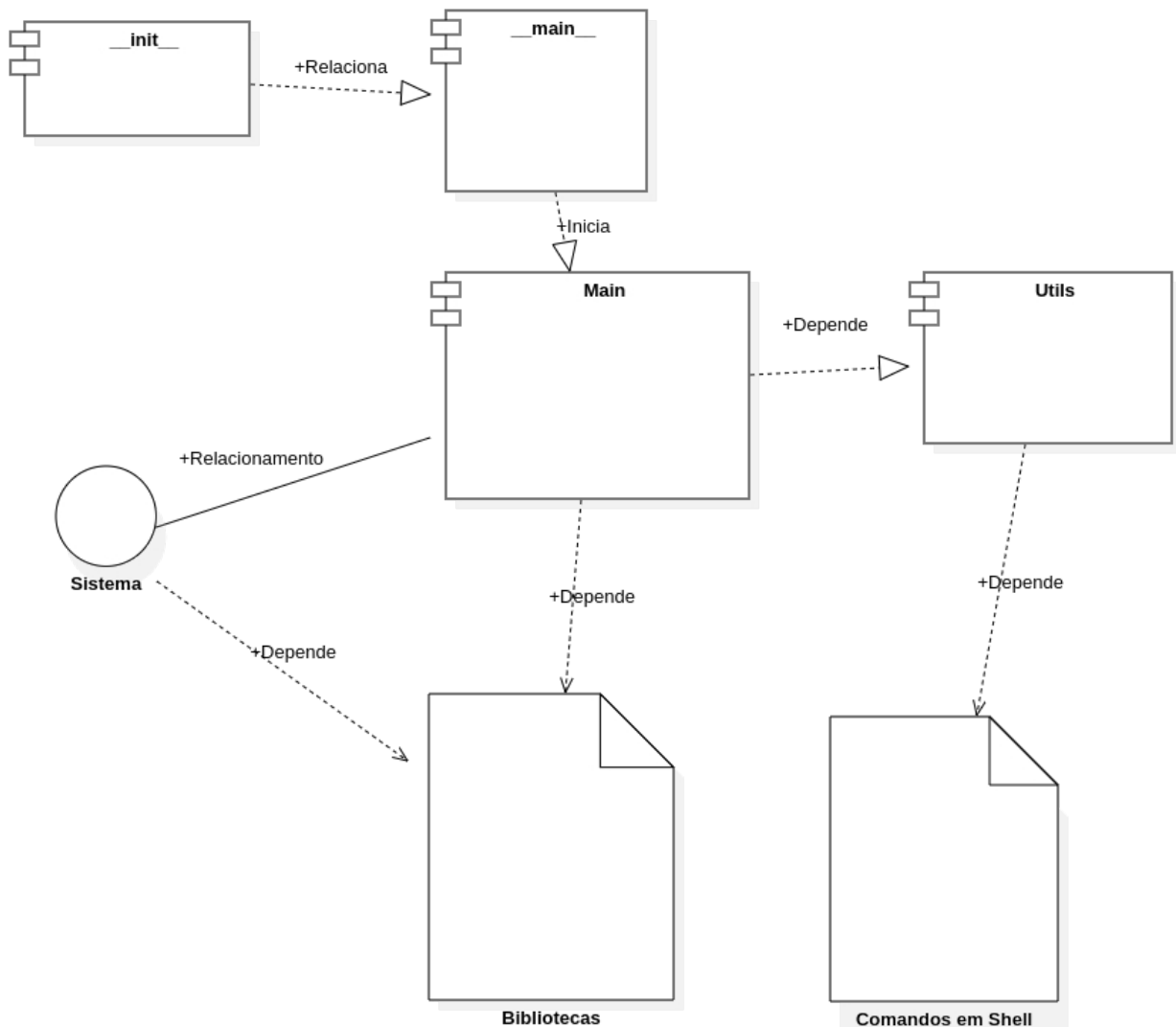
2.2.6 Diagrama de classe



A classe `__init__` iniciará o programa e assim executará o conteúdo da classe `Main`. Em seguida, a assistente virtual ficará no aguardo do comando do usuário para então comunicar-se com a classe `Utils`. Desta forma a classe `Main` invocará a si mesma. A assistente virtual depende dos comandos que vem da classe `Utils` para funcionar. A classe `Utils` chama a si mesma quando necessária.

Nota-se que a classe `Utils` depende da classe `Main` para ser recrutada. Esta integração é feita classificada como classe de agregação da `Main`.

2.2.7 Diagrama de componentes



O componente `__init__` inicia-se por padrão e em seguida este relaciona-se com o `__main__` para então executar o conteúdo de *Main*. O motivo desta redundância deve-se ao modelo padrão do empacotamento do repositório PyPi, o qual hospeda softwares desenvolvidos usando a linguagem de programação Python.

O componente *Main* depende do artefato das bibliotecas necessárias para o funcionamento da assistente virtual. Este componente se relaciona diretamente com o componente *Utils*. O componente *Utils*, por sua vez, depende dos comandos em Shell quando solicitado.

A interface do sistema depende que as bibliotecas do Python estejam instaladas na máquina do usuário para que haja o funcionamento correto.

2.3 DIFICULDADES ENFRENTADAS

2.3.1 Versionamento do Python

Ao decorrer do processo de desenvolvimento do projeto obtivemos dificuldades com relação ao versionamento da linguagem Python. Tal inconveniência ocorreu por conta das funções providas pelas API das bibliotecas ficarem defasadas em versões superior a 3.6 da linguagem. A biblioteca Pyttsx3 foi a mais afetada, e teve que ser substituída pelo software Festival.

2.3.2 Ambientes virtuais

Antes da substituição da biblioteca Pyttsx3 pelo software Festival, tentamos fixar a versão 3.6 da linguagem Python por meio de ambientes virtuais. Tanto o PyENV quanto o ASDF foram testados para este fim, entretanto nunca foi possibilitado o comportamento adequado para a assistente virtual, além de ser um obstáculo adicional a implementação do projeto.

2.3.3 A solução

Em busca de resolver os problemas causados pelo versionamento da linguagem Python, em dado momento do desenvolvimento do projeto, decidimos refatorar o código fonte e implementar a biblioteca Shell Script Library para realizar a comunicação de forma simplificada entre o código fonte escrito em Python e os comandos do terminal do sistema operacional Linux.

3 RESULTADOS

3.1 PRECISÃO DO RECONHECIMENTO DE VOZ

Devido a utilização da biblioteca de reconhecimento de voz Speech Recognition estar em processo de evolução, nem sempre os comandos são detectados conforme o esperado. Em dados momentos, são entendidos comandos com sonoridade parecida, porém cuja o significado seja diferente daquele desejado.

Na tentativa de minimizar este problema, foi criada uma estratégia que armazena um conjunto de palavras similares para cada termo. Desta forma, a assistente virtual ao ser solicitada, identifica se o comando entendido está contido nesta lista de termos similares e assim aumenta a sua precisão.

3.2 EXECUÇÃO DA ASSISTENTE VIRTUAL

A assistente virtual mostrou-se capaz de abrir e fechar programas instalados no sistema operacional; focar em janelas na tela; mover o posicionamento das janelas; mover as janelas para diferentes áreas de trabalho; e, acessar diferentes áreas de trabalho.

Os comandos referentes a controle de janelas funciona exclusivamente no i3WM. Os comandos de abertura de softwares funcionam independentemente do gerenciador de janelas utilizado.

3.3 LISTA DE COMANDOS SUPORTADOS

3.3.1 Ação de abertura de softwares

1. *Software Firefox* abre o navegador Firefox.
2. *Software menu* abre o menu Rofi launcher.
3. *Software Spotify* abre o programa do Spotify.

4. *Software terminal* abre o emulador de terminal padrão do sistema operacional.

3.3.2 Ação do sistema de áudio

1. *System volume up* aumenta o nível de volume do sistema de áudio da sistema operacional.
2. *System volume down* diminui o nível de volume do sistema de áudio da sistema operacional.

3.3.3 Ação do sistema de brilho da tela

1. *System brightness up* aumenta o nível de brilho da tela.
2. *System brightness down* diminui o nível de brilho da tela.

3.3.4 Ação de controle da janela

1. *System window close* fecha a janela da aplicação focada.
2. *System window float* alterna entre o modo de visualização das janelas entre normal e flutuação.
3. *System window full* alterna a janela da aplicação atualmente focada entre normal e tela cheia.
4. *System window stick* alterna o modo de visualização das janelas entre normal e pilha.
5. *System window focus up* muda o foco da janela da aplicação para a janela que está na parte superior da tela.
6. *System window focus left* muda o foco da janela da aplicação para a janela que está na parte esquerda da tela.
7. *System window focus right* muda o foco da janela da aplicação para a janela que está na parte direita da tela.
8. *System window focus down* muda o foco da janela da aplicação para a janela que está na parte inferior da tela.
9. *System window move up* movimenta a janela da aplicação atualmente focada para a parte superior da tela.
10. *System window move left* movimenta a janela da aplicação atualmente focada para a parte esquerda da tela.
11. *System window move right* movimenta a janela da aplicação atualmente focada para a parte direita da tela.
12. *System window move down* movimenta a janela da aplicação atualmente focada para a parte inferior da tela.
13. *System workspace go last* retorna para a última área de trabalho utilizada.
14. *System workspace go <número>* vai para a área de trabalho do valor numérico informado.
15. *System workspace go next* vai para a área de trabalho seguinte.
16. *System workspace go prev* vai para a área de trabalho anterior.
17. *System workspace move last* move a janela da aplicação atualmente focada para a última área de trabalho utilizada.
18. *System workspace move <número>* move a janela da aplicação atualmente focada para a área de trabalho do valor numérico informado.
19. *System workspace move next* move a janela da aplicação atualmente focada para a área de trabalho seguinte.
20. *System workspace move prev* move a janela da aplicação atualmente focada para a área de trabalho anterior.

4 DISCUSSÃO ACADÊMICA

Para os autores Sabino e Kon (2009), o software livre permite que modificações no código fonte sejam realizadas desde que sejam respeitadas as licenças de uso definidas pelo autor do projeto. Isto abre a possibilidade para que os softwares possam ser adaptados para diferentes finalidades, incluindo a acessibilidade.

Indo de encontro, Ball e Buff (2004) afirmam que a licença do código fonte do Linux é permissiva a modificações, pois encontra-se sob os termos GNU. Corroborando com esta premissa, Campos (2006) aponta que a maioria dos softwares incluídos nas distribuições Linux são livres para modificações. Sendo assim, tanto o kernel, as distribuições Linux e muitos dos softwares inclusos podem ser modelados visando a implementação de recursos voltados a acessibilidade sem violar os termos de uso.

Bradley (2018) e Stapelberg (2012) apontam que o gerenciador de janelas i3WM é capaz de dividir o layout de organização das janelas de aplicações tanto de forma horizontal, quanto vertical além de ser controlado via socket UNIX para realizar a comunicação entre o comando requerido e a tecla de atalho configurada. O próprio página oficial do projeto i3 (2013) deixa claro que o i3WM está sob licença BSD, o qual também é uma licença livre. Isto permite que o gerenciador de janelas possa ser modificado para quaisquer fins, inclusive implementando recursos que possam ser direcionados a acessibilidade.

Dentre as bibliotecas e softwares utilizados para construção da assistente virtual VAL, o SpeechRecognition e o Shell Script Library estão sob licença de uso BSD, enquanto que o Festival Speech Synthesis System está sob os termos da licença X11-Type, o qual também é permissiva a modificações (SpeechRecognition, 2019), (BECK, 2022) e (CSTR, 2008).

A assistente virtual VAL está sob licença GPL3, o qual permite que modificações sejam realizadas, porém, as mesmas devem ser disponibilizadas publicamente. Sendo assim, este projeto tem a possibilidade de receber contribuições no código fonte e modelar-se para atender inúmeros objetivos, inclusive a acessibilidade.

5 CONCLUSÃO

A VAL é uma assistente virtual que visa se tornar uma prova de conceitos a respeito da exploração de novas funcionalidades sobre como uma assistente virtual pode atuar no campo da manipulação de comandos relacionados ao controle da interface gráfica por meio do gerenciador de janelas.

A estratégia por trás da utilização do i3WM deve-se sobre como os gerenciadores de janela lado a lado (*tiling window manager*) são projetados por padrão para auto-organizarem as aplicações. Além de redimensionar e reposicionar as áreas visíveis do software na tela, estes gerenciadores de janelas também são capazes de obedecerem comandos via tecla de atalhos para assim efetuar modificações em seu comportamento.

Se por um lado utilizar o mouse ou decorar diversas teclas de atalhos podem ser empecílos para alguns usuários controlarem a interface gráfica do sistema operacional, o projeto VAL proporciona uma nova perspectiva, na qual expande-se o horizonte em favor da acessibilidade, sobretudo, em um cenário que pode beneficiar pessoas com mobilidade reduzida.

A união entre uma assistente virtual e o *tiling window manager* trás resultados com potencial muito promissor e que devem ser explorados em trabalhos futuros. Esta é uma estratégia que pode ter como objetivo final o controle total da interface gráfica de um sistema operacional sem a necessidade da utilização das mãos do usuário para executar tarefas em softwares que não necessariamente foram desenvolvidos para este público em questão.

O legado que este projeto procura deixar é que o desenvolvimento de qualquer que seja a função de um software, desde que haja a possibilidade de ser executado por meio da interface de

linha de comando do terminal do sistema operacional, este recurso poderá ser executado por meio de comandos de voz, uma vez que devidamente programado.

6 APÊNDICE

6.1 PRIORIDADES

	Prioridade
Valor	Descrição
1	Nem um pouco importante
2	Ligeiramente importante
3	Importante
4	Muito importante
5	Extremamente importante

6.2 CATEGORIAS

	Categorias
Valor	Nome
1	Interação com o sistema
2	Funcionalidades
3	Reconhecimento de voz
4	Fala do sistema
5	Registro do sistema
6	Configurações do sistema

6.3 HISTÓRIAS DE USUÁRIOS

Histórias de usuários					
ID	Persona	Planning poker	Prioridade	Categorias	Descrição
HU-01	#UB	Extremamente difícil	5	1	Gostaria que a assistente virtual reconhecesse meus comandos de voz
HU-02	#S	Difícil	1	3	Gostaria que a assistente virtual compreendesse o idioma inglês
HU-03	#S	Difícil	4	3	Gostaria que a assistente virtual compreendesse o idioma português
HU-04	#S	Difícil	2	2	Gostaria que a assistente virtual fosse capaz de abrir programas do computador
HU-05	#S	Fácil	3	2	Gostaria que a assistente virtual fosse capaz de fechar programas do computador
HU-06	#S	Normal	3	5	Gostaria que a assistente virtual registrasse suas ações em um arquivo de log
HU-07	#S	Normal	4	2	Gostaria que a assistente virtual fosse capaz de controlar as janelas da interface gráfica
HU-08	#S	Difícil	3	2	Gostaria que a assistente virtual fosse capaz de controlar o nível de luminosidade da tela do computador
HU-09	#S	Difícil	3	2	Gostaria que a assistente virtual fosse capaz de controlar o volume da saída de áudio do sistema operacional
HU-10	#S	Normal	5	1	Gostaria que a assistente virtual interpretasse os comandos de voz apenas quando ela for chamada
HU-11	#S	Fácil	2	6	Gostaria que a assistente virtual pudesse carregar as informações de sua configuração a partir de um arquivo externo
HU-12	#S	Normal	3	4	Gostaria que a assistente virtual me informasse caso o comando seja inválido
HU-13	#UB	Difícil	2	1	Gostaria de personalizar os comandos da assistente virtual
HU-14	#S	Normal	3	5	Gostaria que a assistente virtual gravasse a solicitação do último comando de voz
HU-15	#S	Difícil	4	3	Gostaria que a assistente virtual compreendesse os comandos mesmo quando offline
HU-16	#S	Normal	2	6	Gostaria que a assistente virtual não solicitasse permissões de privacidade do usuário

6.4 REGRAS DE NEGÓCIO

ID	Regras de negócio Descrição
RN- 01	Usuário deverá ter acesso ao sistema sem a necessidade de privilégios de administrador
RN- 02	O usuário não precisará ter permissão de administrador do sistema para executar a aplicação
RN- 03	O usuário deverá editar o arquivo de texto correspondente as configurações do sistema caso deseje realizar customizá-lo
RN- 04	O sistema não disponibilizará interface gráfica ao usuário
RN- 05	O usuário deverá ser informado caso um comando seja invalidado pelo sistema
RN- 06	O usuário deverá estar ciente que a utilização do software será de sua inteira responsabilidade
RN- 07	Os comandos validados pela assistente virtual deverão ser executadas através do terminal Linux
RN- 08	Deverá ser reconhecido pelo sistema comandos no idioma inglês
RN- 09	Deverá ser reconhecido pelo sistema comandos no idioma português
RN- 10	A assistente virtual deverá ser possibilitada de customização
RN- 11	A assistente virtual deverá reconhecer comandos de voz emitidos pelo usuário
RN- 12	A assistente virtual só estará em execução de segundo plano
RN- 13	A assistente virtual deverá ser ativada apenas quando chamada através de ser codinome (padrão: VAL)
RN- 14	Deverá ser mantido pelo usuário a gravação apenas do seu último comando
RN- 15	Deverá ser criado pela assistente virtual um arquivo de log de funcionamento do sistema
RN- 16	Deverá ser carregado as configurações do sistema através de um arquivo externo
RN- 17	Deverá ser reconhecido os comandos de voz mesmo em ambiente de conexão offline

6.5 REQUISITOS FUNCIONAIS

Requisitos Funcionais		Histórias de usuários	Regras de negócio
ID	Descrição		
RF-01	O usuário requisitará uma funcionalidade ao sistema	HU-02	RN- 13
RF-02	O sistema deverá registrar arquivos de logs	HU-07	RN- 15
RF-03	O sistema deverá gravar a voz do usuário	HU-15	RN- 14
RF-04	O sistema deverá permitir apenas o uso de funcionalidades disponíveis	HU-13	RN- 05
RF-05	O sistema controlará as janelas da interface gráfica	HU-08	RN- 07
RF-06	O sistema deverá reconhecer a voz do usuário offline	HU-16	RN- 17
RF-07	As bibliotecas requeridas para o funcionamento do sistema deverão armazenar os idiomas inglês	HU-03	RN- 08
RF-08	As bibliotecas requeridas para o funcionamento do sistema deverão armazenar os idiomas português	HU-04	RN- 09
RF-09	O sistema deverá carregar suas configurações através de um arquivo externo de notações de objetos	HU-12	RN- 16
RF-10	O sistema deverá permitir que o usuário crie e carregue configurações personalizadas	HU-14	RN- 03

6.6 REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais		Regras de negócio	Histórias de usuários
ID	Descrição		
RNF – 01	Apenas a última gravação da voz do usuário deverá ser mantida no sistema	RN- 14	HU-15
RNF – 02	A biblioteca do idioma inglês será disponibilizada	RN- 08	HU-03
RNF – 03	A biblioteca do idioma português será disponibilizada	RN- 09	HU-04
RNF – 04	Não haverá permissões de privacidade deverão a serem solicitadas	RN- 01	HU-16
RNF – 05	A assistente de voz deverá informar o usuário caso o comando solicitado seja inválido através de emissão sonora por meio de sua própria fala	RN- 05	HU-13
RNF – 06	A assistente de voz deverá informar o usuário caso o comando solicitado seja inválido através de texto por meio da notificação do sistema	RN- 05	HU-13

7 REFERÊNCIAS BIBLIOGRÁFICAS

ALEXANDRA Ap. Marcelino Toscaro, CLÁUDIO Martinez, LUCIANA Meneghel, MARCELO Araújo, FRANCO (Coordenador do projeto e organizador da publicação), RENATA A. Fonseca del Castillo, ROANDER Scherrer, RUBENS Queiroz de Almeida (Coordenador da DSC-CCUEC). Preparação de Conteúdo para WEB. 2003. EAD- Unicamp. 62f.

BALL, B.; DUFF, H. Dominando Linux Red Hat e Fedora: Conhecimento, Soluções, Especialização. São Paulo: Makron Books. 2004.

BECK, Henrik; 2022. Disponível em: <https://github.com/henrikbeck95/shell_script_library>. Acesso em: 14 junho 2022.

BRADLEY, Evan. An Infinite-Pane, Zooming User Interface Window Manager and Survey of X Window Managers. 2018. Tese de Doutorado.

CAMPOS, Augusto. O que é uma distribuição Linux. **BR-Linux. Florianópolis**, 2006.

CARDOSO, S. H. Tecnologia de reconhecimento de voz. 2001. Disponível em: <<http://www.edumed.org.br/cursos/biblioteca/recognition1.html>>. Acesso em: 2 maio 2016.

CASTRO, Jose Dieguez. Arch linux. In: **Introducing Linux Distros**. Apress, Berkeley, CA, 2016. p. 235-252.

CSTR; 2008. Disponível em: <<https://www.cstr.ed.ac.uk/projects/festival/>>. Acesso em: 18 junho 2022.

CYBIS, Walter de Abreu. A identificação dos objetos de interfaces home-computador em seus atributos ergonômicos. Tese de Doutorado em Engenharia de Produção. Programa de Pós-Graduação em Engenharia de Produção, UFSC, Florianópolis, 1994.

DOS SANTOS, Carlos Roberto; BERNAL, Janaína Cristina Galvez; FERNANDEZ, Breno Ortega. AUTOMAÇÃO RESIDENCIAL POR COMANDO DE VOZ PARA VIABILIZAR A ACESSIBILIDADE DE PESSOAS COM DEFICIÊNCIA. Revista Estudos & Pesquisas Unilins, v. 1, n. 1, p. 49-55, 2018.

IBM; 2021. Disponível em: <https://www.ibm.com/br-pt/cloud/watson-speech-to-text?utm_content=SRCWW&p1=Search&p4=43700052743482493&p5=e&gclid=CjwKCAiA-9uNBhBTEiwAN3iINFlCmtI-B9kCiVIYsedE4ngg7iTsWoLluEEerpiYT1S41DlZSljJ-hoCnIkQAvD_BwE&gclsrc=aw.ds>. Acesso em: 13 dezembro 2021.

IBGE. Censo Demográfico 2010 - Características gerais da população, religião e pessoas com deficiência - Características urbanísticas do entorno dos domicílios. Rio de Janeiro: cap. 1.3. Instituto Brasileiro de Geografia e Estatística - IBGE, 2012.

GOUVÊA, Bernardo Andrade; JAMIL, George Leal. Linux: Uma análise histórica mercadológica e funcional de um sistema operacional. Revista Pretexto, 2002.

I3; 2013. Disponível em <<https://i3wm.org/>>. Acesso em: 14 junho 2022.

ITS. Tecnologia Assistiva nas Escolas - Recursos básicos de acessibilidades sociodigital para pessoas com deficiência. São Paulo: cap. 1 e 2. Instituto de Tecnologia Social (ITS Brasil), 2008.

PYTTSX3; 2017. Disponível em: <<https://pyttsx3.readthedocs.io/en/latest/>>. Acesso em: 04 junho 2022.

SABINO, V. C. Um estudo sistemático de licenças de software livre. Dissertação apresentada ao instituto de matemática e estatística da Universidade de São Paulo (USP). São Paulo, 2011.

SABINO, Vanessa; KON, Fabio. Licenças de software livre história e características. 2009.

SPEECHRECOGNITION; 2019. Disponível em:
<https://github.com/Uberi/speech_recognition>. Acesso em: 13 dezembro 2021.

SPHINX4; 2014. Disponível em: <<https://github.com/cmusphinx/sphinx4>>. Acesso em: 13 dezembro 2021.

STAPELBERG, Michael. i3-msg(1). <https://build.i3wm.org/docs/i3-msg.html>. 2012.