**TGI-II 2022 em**

**VAL – ASSISTENTE VIRTUAL LINUX**

**Bruno M. S. Cerqueira, Jeanderson C. P. Sardinha, Matheus A. R. Cunha**

Universidade Cruzeiro do Sul

Av. Dr. Ussiel Cirílo, 225 - Vila Jacuí, São Paulo - SP, 08060-070 - Brasil

vhvic

**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 multimidia por meio da interação por voz são alguns dos benefícios a serem desfrutados deste software.

**Palavras-chaves:** VAL, Python, Yaml, PyAudio e Linux.

**Abstract**

The development and configuration of a virtual assistant is a means of providing an integration tool between the operating system and humans, a scenario in which people with reduced mobility can benefit. Providing a wider range of possibilities regarding software accessibility tools can make the user decide which one is best for his individuality. The existence of resources 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.

**Keyword:** VAL, Python, Yaml, PyAudio e Linux.

**1. Introdução**

**1.1 Apresentação do tema**

Conforme o numero maior de pessoas com problemas de comorbidades vem aumentando a necessidade da tecnologia vem como isto vem aumentando ano após anos varias pessoas hoje em dia não consegue navegar no Windows ou no Linux de maneira pratica, é por isso que estamos aqui desenvolvendo a nossa assistente virtual que poderá ajudar essas pessoas com praticidade e efetividade.  
Onde a pessoa não precisara de clicar ao iniciar um processo no sistema, a VAL fara todo o trabalho ao usuário assim tornando a vida mais pratica e fácil. A VAL é uma assistente virtual que fara com que o sistema inicie programas, faça algumas tarefas que estão listadas no manual dela que estará descrito nos próximos itens.

**1.2 Problema de pesquisa**

Ao decorrer do processo de pesquisa sobre as tecnologias necessárias para criação da assistente de voz, pôde-se identificar a existência de algumas bibliotecas de código que permitissem o reconhecimento e a transcrição da voz humana no computador. Escolher quais destas bibliotecas utilizar implacaria tanto no licenciamento do projeto; no processo de desenvolvimento do código fonte, e sobretudo, no processo de criação da engenheria de software. Ao avaliarmos tais bibliotecas, escolhemos a Speech Recognition por conta da sua licença de uso e estar recebendo atualização dos desenvolvedores do projeto. Estes critérios foram fundamentais para exclusão das demais alternativas, tais como a Watson Speech to Text (IBM - Python) e o Sphinx4 (Java), além de direcionar a códificação do projeto para a linguagem Python.

De imediato, demos início ao desenvolvimento do projeto apenas em caráter experimental visando conhecer e explorar as possibilidades que poderíamos utilizar. Após um longo período de sofrimento conseguimos identificar as dependencias necessárias para o funcionamento da VAL. A dificildade mencionada deve-se a incompatibilidade de versionamento entre o PyAudio e o Speech Recognition em versões superiores a 3.6 do Python.

Utilizamos o sistema operacional ArchLinux como ambiente de testes e isto deve-se aos seguintes motivos: familiaridade de um dos integrantes do grupo ter com este sistema operacional; licença de uso do sistema operacional em questão; o sistema operacional possuir uma vasta documentação; e por fim, a organização do gerenciamento de pacotes de programas, sendo claramente dividido entre repositórios disponibilizados pelos mantenedores oficiais do projeto e os softwares disponibilizados pelos demais usuários, a Arch User Repository (AUR), a qual poderia facilitar nos processos de busca e instalação de dependencias dos softwares.

**1.3 Objetivo**

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.

**1.4 Justificativa**

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 multimidia por meio da interação por voz são alguns dos benefícios a serem desfrutados deste software.

**2. Material e Métodos**

Inicialmente, construímos a concepção do projeto em base de artigos científicos, os quais tratam do mesmo assunto/objetivo que estamos propondo neste trabalho. Abordando a eficácia referente ao uso de Big Data em programas de gestão hospitalar, que engloba as perspectivas futuras da utilização dele no Brasil, o artigo de Alexandre D. P. C. Filho “Uso de big data em saúde no Brasil: perspectivas para um futuro próximo” presente em: <https://www.scielosp.org/article/ress/2015.v24n2/325-332/> faz uma elaboração das três principais áreas propensas ao uso desde tipo de estrutura de dados na área da saúde: medicina de precisão, prontuários eletrônicos do paciente e internet das coisas. Contudo, além da perspectiva futura de sua utilização, precisávamos buscar informações do seu uso de fato no mundo corporativo, com detalhes, para revisar as oportunidades e desafios que este tipo de implementação poderá trazer.

Visando tal discussão, encontramos um artigo que constrói a ideia da revisão de sua utilização no mundo corporativo. Tal artigo foi feito pela empresa Internacional Symposium on Project Management, Innovation and Sustainability, disponível em: <https://singep.org.br/4singep/resultado/245.pdf>.

Porém o que nos cativou a desenvolver um projeto baseado em um banco de dados não relacional foi a diferença de armazenamento e velocidade na recuperação de dados deste tipo de estrutura comparada com a estrutura relacional. Para o nosso protótipo, faremos testes visando os resultados apresentados pelo artigo de Marcos A. P. M. Filho, disponível em <https://www.cin.ufpe.br/~tg/2014-2/mapmf.pdf>, que realiza a análise de desempenho estre os dois tipos de banco de dados.

Tendo em mente os resultados obtidos nos artigos científicos citados anteriormente, nosso projeto visa por meio desta estrutura comprovar a eficácia de armazenamento e de velocidade do banco de dados não-relacional aplicado no sistema de saúde.

Para expressarmos um resultado adequado ao que estamos propondo, faz-se necessário o uso dos dois tipos de banco de dados (Relacional e não-relacional) para analisar e comparar a performance de um sistema usando cada um deles.

Para provar a efetividade do modelo não-relacional e para atingir o objetivo proposto, começaremos pela demonstração do modelo (relacional) que consideramos menos eficaz para esse tipo de sistema. O desenvolvimento do projeto que o utilizará, se baseará na linguagem de programação Java com a implementação de Servlets, classes que atuarão como gerenciadores de requisições e respostas entre o cliente e o servidor. Haverá uma Servlet gerencial (principal) que funcionará como filtro indicando para qual classe as requisições feitas deverão ser encaminhadas. Esta classe que recebe a requisição, implementará os métodos básicos para conseguir efetuar o tratamento da requisição e conseguir devolver uma resposta adequada para tal (Métodos doGet e doPost, que são nativos e obrigatórios para a implementação da Servlet).

A programação Java fará a “comunicação” com o usuário através de páginas JSP (JavaServer Pages) que implementarão a interface gráfica para o usuário poder interagir com o sistema, as quais serão construídas em HTML5, CSS3, Bootstrap4 e AJAX, e por fim farão contato com as classes gerenciais através da programação Javascript. O algoritmo das classes gerenciais, terão como objetivo “traduzir” a informação recebida através da requisição feita pelo Javascript e fazer contato com o banco de dados pelo modelo DAO, que utilizará JDBC para conseguir fazer essa ponte entre os diferentes tipos de ambiente.

O banco de dados será construído em uma linguagem de programação estruturada, conhecida formalmente como SQL e de forma geral este programa utilizará o modelo MVC em seu desenvolvimento, o qual funciona da seguinte forma: uma classe gerencial recebe a requisição do usuário, ela chama os métodos subsequentes podendo, ou não, realizar o contato com o banco de dados através de métodos DAO e retorna ao usuário uma resposta adequada para o que lhe foi pedido.

Para a demonstração com uso de banco de dados não relacional analisamos dentre várias opções; MongoDB, Redis, Cassandra, entre outras. Escolhemos MongoDB por ser um banco extremamente popular e em estado maduro, bem como sua estrutura baseada em Documents que são muito semelhantes a arquivos JSON o que permite uma flexibilidade maior com a forma com que persistimos os dados, o que consideramos fundamental para o setor hospitalar e de prontuários digitais.

Optamos por criar um sistema de CRUD que simula as tarefas cotidianas do domínio hospitalar; cadastro de pacientes, registro de atendimentos, efetuar internações e obter listas de pacientes internados. Utilizamos a linguagem javascript com o uso da tecnologia Nodejs para criar o nosso backend, auxiliado pelo framework Express para criar a aplicação Web. A escolha foi motivada por conta da Stack de desenvolvimento Nodejs, Express e MongoDB ser bastante popular e difundida.

**3. Resultados**

As atividades referentes ao trabalho de graduação interdisciplinar foram iniciadas objetivando a construção de um sistema operacional baseado em Linux voltado para pessoas com mobilidade reduzida. Todavia ao refletimos sobre a necessidade de criação do desenvolvimento deste sistema operacional, percebemos que o projeto teria maior potencial de adesão para o público alvo pretendido caso não fosse requerido à instalação do sistema operacional por completo, mas em seu lugar apenas a assistente de voz, que por si só seria o coração do projeto. Esta decisão permite que o usuário final possa utilizar as mesmas funcionalidades não importando a distribuição Linux instalada, desde que estejam utilizando a interface gráfica i3.

Uma vez que esta importante decisão foi tomada, as atividades referentes ao trabalho de graduação interdisciplinar foram alteradas para objetivar a construção de uma assistente de voz que fosse capaz de controlar a interface gráfica i3 no Linux. O projeto foi nomeado de VAL - Voice Assistant for Linux -, ou em tradução livre, Assistente de Voz para Linux.

No início do processo de criação nos propusemos a pesquisar sobre as tecnologias existentes para que pudéssemos compreender o cenário em que pretendíamos atuar. Para nossa surpresa, apesar de algumas assistentes virtuais existirem, nenhuma delas continha a funcionalidade de controlar as janelas da interface gráfica, ficando restrita apenas a abrir determinados programas e/ou algumas funcionalidades extras de acordo com a implementação de cada desenvolvedor.

Iniciamos a pesquisa do projeto visando descobrir quais as tecnologias que seriam necessárias para criação da assistente de voz. Identificamos a existência de algumas bibliotecas que permitem o reconhecimento e a transcrição da voz humana no computador sendo as mais interessantes: Watson Speech to Text (IBM - Python), Sphinx4 (Java) e Speech Recognition (Python). Optamos pela biblioteca Speech Recognition por conta da sua licença de uso e também por ainda estar recebendo atualizações dos desenvolvedores do projeto e isto nos direcionou a utilizarmos como linguagem de programação o Python.

Em um primeiro momento, o desenvolvimento do projeto foi iniciado apenas para criarmos alguns comandos experimentais com o propósito de analisarmos o comportamento, a viabilidade do desenvolvimento e a operabilidade das funções provindas das API das bibliotecas necessárias para o funcionamento da VAL. Neste aspecto vale a pena ressaltar que a parte mais trabalhosa foi a instalação destas bibliotecas em ambiente Linux devido a sua incompatibilidade de versionamento entre o PyAudio e o Speech Recognition em versões superiores a 3.6 do Python.

Ao superar o obstáculo das bibliotecas, o qual levou mais de dois meses para ser contornado, realizamos alguns testes de implementações no sistema operacional Arch Linux que se mostraram satisfatórios e muito promissores. Apesar de haver alguns problemas com relação a precisão da detecção do reconhecimento de voz, pudemos elaborar estratégias para que a assistente virtual detectasse comandos tanto no idioma inglês quanto em português desde que previamente alterada esta configuração no arquivo específico para tal. Em testes iniciais a VAL se mostrou capaz de abrir e fechar programas; focar em janelas na tela; mover o posicionamento das janelas; mover as janelas para diferentes áreas de trabalho; e, acessar diferentes áreas de trabalho.

Ao entender o quê o projeto precisaria para funcionar, sejam as suas dependências de desenvolvimento e também a sua lógica computacional, demos início ao planejamento de sua engenharia de software que segue em andamento. Uma vez que a engenharia de software do primeiro ciclo de desenvolvimento da assistente virtual estiver concluída pretendemos refatorar o código fonte anteriormente esboçado e assim darmos continuidade ao seu desenvolvimento tendo desta vez um panorama mais claro do projeto como um todo.

Após definirmos que o objetivo do trabalho de graduação interdisciplinar consistiria na construção de uma assistente virtual para controlar o gerenciador de janelas i3WM no Linux através de comandos por voz, iniciamos uma pesquisa sobre as tecnologias existentes com o intuito de assimilar e distinguir o cenário em que pretenderíamos atuar.

A escolha de um nome apropriado e condizente com a descrição do projeto (A VAL - Voice Assistant for Linux -, ou em tradução livre, Assistente de Voz para Linux), tem nos direcionado a delimitar o escopo a ser trabalhado. Tal definição nos propricia flexibilidade para implementação de funcionalidades.

Ao decorrer do processo de pesquisa sobre as tecnologias necessárias para criação da assistente de voz, pôde-se identificar a existência de algumas bibliotecas de código que permitissem o reconhecimento e a transcrição da voz humana no computador. Escolher quais destas bibliotecas utilizar implacaria tanto no licenciamento do projeto; no processo de desenvolvimento do código fonte, e sobretudo, no processo de criação da engenheria de software. Ao avaliarmos tais bibliotecas, escolhemos a Speech Recognition por conta da sua licença de uso e estar recebendo atualização dos desenvolvedores do projeto. Estes critérios foram fundamentais para exclusão das demais alternativas, tais como a Watson Speech to Text (IBM - Python) e o Sphinx4 (Java), além de direcionar a códificação do projeto para a linguagem Python.

De imediato, demos início ao desenvolvimento do projeto apenas em caráter experimental visando conhecer e explorar as possibilidades que poderíamos utilizar. Após um longo período de sofrimento conseguimos identificar as dependencias necessárias para o funcionamento da VAL. A dificildade mencionada deve-se a incompatibilidade de versionamento entre o PyAudio e o Speech Recognition em versões superiores a 3.6 do Python.

Utilizamos o sistema operacional ArchLinux como ambiente de testes e isto deve-se aos seguintes motivos: familiaridade de um dos integrantes do grupo ter com este sistema operacional; licença de uso do sistema operacional em questão; o sistema operacional possuir uma vasta documentação; e por fim, a organização do gerenciamento de pacotes de programas, sendo claramente dividido entre repositórios disponibilizados pelos mantenedores oficiais do projeto e os softwares disponibilizados pelos demais usuários, a Arch User Repository (AUR), a qual poderia facilitar nos processos de busca e instalação de dependencias dos softwares.

Superar estes obstáculos e entender os motivos pelos quais diversas mensagens de erros surgiram durante todo este processo, nos condicionou a ter uma visão do comportamento do software. Identificamos possíveis limitações das bibliotecas quanto a detecção do reconhecimento de voz e também da pronúncia provinda a partir de texto no ambiente Linux ser um pouco robotizada. Todo este conhecimento adquirido nos ajudou a iniciarmos o processo de engenharia de software, onde estamos nos empenhando atualmente. Simplificar e organizar através de documentação é de fundamental importancia para que possamos reiniciar o processo de codicação de forma muito mais objetiva, uma vez que a VAL se mostrou capaz de abrir e fechar programas; focar em janelas na tela; mover o posicionamento das janelas; mover as janelas para diferentes áreas de trabalho; e, acessar diferentes áreas de trabalho.

Software em uso Demonstração: Video do youtube

**4. Discussão**

Com base nos resultados apresentados anteriormente, Obtivemos sucessos em nossas pesquisas para melhorar o software, o aperfeiçoamos dos tempos pra ca e aprimoramos ele assim concertando muitos defeitos apresentados no caminho de nossa jornada, hoje em dia o software apresenta uma melhora grande das execuções dos comandos porem ainda necessita do usuário pronunciar corretamente o comando.

Com base no que foi apresentado no TGI-I o projeto teve um avanço significativo na melhora

//TERMINAR DISCUSSAO

//FAZER CONCLUSAO

**6. Referências bibliográficas**

André Pereira Martins Filho, Marcos. SQL X NOSQL: ANÁLISE DE DESEMPENHO DO USO DO

MONGODB EM RELAÇÃO AO USO DO POSTGRESQL. Universidade Federal de Pernambuco, fev. de 2015. Disponível em: <https://www.cin.ufpe.br/~tg/2014-2/mapmf.pdf>. Acesso em: 15 de jun. de 2021.

Barbara de Oliveira, Saulo. O PLANEJAMENTO ESTRATÉGICO E A VISÃO BASEADA EM RECURSOS (RBV). RECADM, São Paulo SP - Brasil, 2013. Disponível em: <https://doaj.org/article/f956be0b766e463b9cd1470602409223>. Acesso em: 15 de jun. de 2021.

Diego Carlos Farias e Fernando Oliveira de Araújo. Gestão hospitalar no Brasil: revisão da literatura visando ao aprimoramento das práticas administrativas em hospitais, jun. de 2017. Disponível em: [https://www.scielosp.org/article/csc/2017.v22n6/1895-1904/pt/#](https://www.scielosp.org/article/csc/2017.v22n6/1895-1904/pt/). Acesso em: 16 de out. de 2021.

Silveira, Márcio. O BIG DATA E SEU USO CORPORATIVO: UMA REVISÃO DE LITERATURA. SINGEP, out. de 2015. Disponível em: <https://singep.org.br/4singep/resultado/245.pdf>. Acesso em: 15 de jun. de 2021.

Machline, Claude. Rede hospitalar nacional usa indicadores gerenciais na

administração de suas unidades. O Mundo da Saúde, jan. de 2011. Disponível em: <https://bvsms.saude.gov.br/bvs/artigos/rede_hospitalar_nacional_usa_indicadores_gerenciais.pdf>. Acesso em: 15 de jun. de 2021.

Artur de Souza, Antônio. CONTROLE DE GESTÃO EM ORGANIZAÇÕES HOSPITALARES. Revista de Gestão USP, set. de 2009. Disponível em: <https://www.revistas.usp.br/rege/article/view/36675/39396>. Acesso em: 21 de jun. de 2021.