

Arquitetura Erika: Aplicação em uma Cadeira de Rodas com Navegação e Reconhecimento Social por Meio de um Chatbot

Anderson Leite * Érika B. Correia ** Giovane Fernandes Moreira *
Marcelo Sampaio ** João Vilasboas ** Adriel Bastos **
José Díaz-Amado ** João Soares **

* *Bacharelado em Sistema de Informação, Instituto Federal da Bahia, BA, Vitória da Conquista (andersonleite.bsi@gmail.com, giovane.fern.m@gmail.com).*

** *Engenharia Elétrica, Instituto Federal da Bahia, Vitória da Conquista, BA (erika.brito.correia@gmail.com, mvsampaio98@gmail.com, joaopfvilasboas@gmail.com, adrielluiz35@gmail.com, jose_diaz@ifba.edu.br, joaoerivando@yahoo.com.br)*

Abstract: The coexistence of service robots in social media has intensified, making human-robot interaction increasingly fluid and necessary. In this context, the present work aims to apply an architecture called Erika responsible for integrating a chatbot that interacts by voice and text command with the autonomous navigation of a wheelchair, which respects social restrictions based on the proxemics zones. An Application Programming Interface is responsible for connecting the chatbot with a web application that communicates with Robot Operating System. The results found demonstrate that the wheelchair can respond to the commands provided by the chatbot, so that, finally, it can trigger autonomous or manual navigation, when necessary.

Resumo: A convivência de robôs de serviço em meios sociais tem se intensificado, tornando a interação homem-robô cada vez mais fluída e necessária. Neste contexto, o presente trabalho tem por objetivo a aplicação de uma arquitetura denominada Erika que é responsável por integrar um *chatbot* que interage por comando de voz e texto com a navegação autônoma de uma cadeira de rodas, a qual respeita as restrições sociais baseando-se nas zonas proxêmicas. Uma Interface de Programação de Aplicativos é responsável por conectar o *chatbot* com uma aplicação *web* que por sua vez estabelece comunicação com o Sistema Operacional de Robôs. Os resultados encontrados demonstram que a cadeira de rodas consegue atender aos comandos fornecidos pelo *chatbot*, para que por fim, possa acionar a navegação autônoma ou manual, quando for necessário.

Keywords: Chatbot; API; Wheelchair; Navigation, Interaction.

Palavras-chaves: Chatbot; API; Cadeira de rodas; Navegação; Interação.

1. INTRODUÇÃO

Diante do contexto tecnológico vigente no século XXI, as interações entre homens e máquinas se desenrolam em meios de convívio popular. Com isso, a utilização de sistemas computacionais contribui para a realização de atividades nos diversos setores do trabalho humano, além de auxiliar em tarefas diárias em suas residências. Dessa maneira, a pesquisa em torno da HRI (*Human-Robot Interaction*) (Sheridan, 2016) possibilitou grandes avanços científicos da área de automação em diversos aspectos, principalmente para a integração de robôs de serviço (Schraft, 2000) em ambientes sociais.

* Esta pesquisa foi apoiada pela Chamada Pública 05/2020 - Seleção de projetos de apoio ao empreendedorismo inovador com foco na Economia 4.0, pelo PIBIC/PIBITI.

A implementação de arquiteturas inteligentes com acréscimo das atuais técnicas de aprendizagem de máquina (Wang, 2016) produzem uma realidade muito mais interativa para os seres humanos, com maiores explorações de informação, recursos e conforto. Podemos exemplificar esta nova tendência com a elaboração de meios de conversação autônoma incorporadas em atendimentos de hospitais, hotéis, aeroportos e muito mais, os chamados *chatbots* para atividades de serviço (Rosruen, 2018).

Conforme o trabalho de Adamopoulou e Moussiades (2020), a Inteligência artificial está incrementando o cotidiano digital por meio de *softwares* e *hardwares* sofisticados. Neste contexto, os *chatbots* promovem inúmeras vantagens para os seus usuários, tal como uma comunicação simples, diversificação de idiomas e a não necessidade de um conhecimento técnico para compreender o seu funcionamento.

A tecnologia para a criação de um sistema de interação por voz ou texto pode ser estruturada por alguns meios de processamento, um destes seria o NLP (*Natural Language Processing*) (Lalwani, 2018), técnica de análise que compreende a linguagem humana. Este entendimento tecnológico é benéfico em outras áreas dos robôs sociais, como o exemplo do auxílio para a navegação móvel (Baus, 2002) através dos dados obtidos em conversas com o usuário. Logo há a possibilidade de promover uma interatividade sobre a movimentação do robô e seu destino.

Desta forma, a proposta neste trabalho é a aplicação da arquitetura denominada Erika que possibilita a colaboração entre um *chatbot* e navegação social móvel, para ser implementada em uma cadeira de rodas autônoma. A comunicação se tornou possível por meio da construção de uma página *web* interativa (Qi, 2009).

Com isso, foi construído um sistema capaz de detectar as perguntas do usuário por comando de voz, com o uso de processamento de linguagem de voz. Desta maneira pode-se retornar respostas e executar a navegação autônoma da cadeira de rodas. Contudo, durante o processo de desenvolvimento surgiu a necessidade de integrar o *chatbot* com um *web site* responsável pela comunicação com a cadeira de rodas que se encontra em ambiente de simulação virtual — ROS (*Robot Operating System*) (Koubâa, 2017). Para integrar o *chatbot* ao *web site* é necessário o uso de uma API (*Application Programming Interface*) (Ofoeda, 2019), no que se refere não foi encontrada na literatura nenhuma que atendesse as necessidades do projeto. Diante disso foi criada a arquitetura Erika, sendo a responsável por realizar a interligação entre todos os *softwares*.

Vale ressaltar que a comunicação da aplicação *web* com a navegação autônoma é estabelecida por meio da biblioteca RosLibJS (Karaca, 2020). Além disso, a utilização da ferramenta GProxemic (Vilasboas, 2021) que elabora melhorias no sistema para respeitar as zonas próximas configuradas conforme a necessidade do usuário e o ambiente, visando trazer uma melhora na socialização entre robôs e seres humanos.

O objetivo principal deste artigo é discorrer sobre a utilização de uma arquitetura interativa entre o usuário e uma cadeira de rodas autônoma, por meio do desenvolvimento de uma API que permita a interação de um *chatbot* com o sistema de navegação social desenvolvido no robô protótipo de módulo de *software* capaz de se comunicar com o usuário. O estudo de caso diz respeito a um *Chatbot* usado para ajudar pessoas com deficiência motora a se locomoverem por meio de uma cadeira de rodas autônoma. Após apresentar uma análise sobre o estado da arte na Seção 2, a metodologia abordada no decorrer do projeto será apresentada na Seção 3, serão posteriormente apresentados os resultados obtidos na Seção 4. Por fim, na Seção 5 são feitas as considerações finais.

2. TRABALHOS RELACIONADOS

Como dito anteriormente, uma das questões é estabelecer a comunicação entre todos os *softwares* que compõem o sistema desenvolvido. A abordagem mais encontrada na literatura é o uso de APIs externas disponíveis gratuitamente para melhorar o desempenho dos *chatbots* e estabelecer

a comunicação com diferentes plataformas. Existem um número considerável de trabalhos que optam em integrar os *chatbots* com APIs disponibilizadas pelas plataformas de mensagens instantâneas (Line, Twitter, Facebook e Telegram) (Yusoff, 2017), deste modo hospedando dentro delas o robô de atendimento virtual. A partir deste ponto de vista é possível entender como ocorre esse fluxo de informações.

Em Gomes (2020) pode-se observar o desenvolvimento da ÁGATA, um *chatbot* para a difusão da educação ambiental, para sua construção foi considerado a disponibilidade de APIs já existentes que atendessem as necessidades do projeto, para isso foi integrado o servidor da ÁGATA com o Telegram (Modrzyk, 2018) pois a plataforma oferece essa conexão.

Outro estudo que traz a mesma abordagem, seria o *chatbot* para monitoramento de histórico acadêmico em Instituição de ensino superior. O sistema foi construído para atender solicitações de informações acadêmicas para os pais dos alunos. O sistema envolve vários processos, como usar o *webhook* (Setiaji, 2018) para receber mensagens e usar a API do Bot disponibilizada pelo Telegram para enviar mensagens em resposta (Heryandi, 2020).

Nos últimos tempos, os maiores esforços se concentraram em tentar criar um robô que pudesse ser o mais parecido possível com um humano: o fluxo de diálogo desempenha um papel fundamental na modelagem de um *Chatbot* (Lombardi, 2019). Na construção do robô de navegação autônoma capaz de conversar e escanear a temperatura corporal para ajudar na triagem do COVID-19, também é possível notar a necessidade de consumir API externa — Google Mobile Vision para compor o sistema proposto, bem como o desenvolvimento do *chatbot* pela plataforma do Dialogflow (Kim, 2021).

Quando se fala de programação de robôs envolve muita complexidade para desenvolver *softwares*. O ROS é composto de bibliotecas que possibilita acelerar o desenvolvimento e reduzir a complexidade do *software*. O ROS fornece uma arquitetura distributiva, que permite compartilhamento de informações sem depender de um único *link*. Rosbridge é um servidor *web-socket*, que estabelece uma camada extra para o ROS. Essa ponte permite a comunicação de aplicativos que não usam o ROS com aplicativos ROS. Roslibjs é uma biblioteca ROS baseada em Javascript que é usada em aplicações particularmente usando soquetes IPs, autorizando-os a fazer interface com aplicativos ROS (Khaliq, 2021).

A navegação robótica social em ambientes habitados por humanos é uma área que começa a evoluir considerando a teoria da proxêmica proposta por Hall (Hart, 2002) e cinco dimensões proxêmicas Greenberg (2011): Distância, Identidade, Localização, Movimento e Orientação (DILMO). Tendo isso em vista, os robôs sociais adotam restrições em sua navegação, respeitando assim as regras da proxêmica, tendo como principal interesse evitar a invasão de espaços íntimos ou pessoais de humanos ao percorrer o caminho (Daza, 2021). Considerando o que foi dito anteriormente, (Vilasboas, 2021) desenvolveu um sistema denominado GProxemic Navigation capaz de complementar a navegação autônoma, respeitando o espaço individual

de acordo com o ambiente, aplicando o conceito de zonas próximas na robótica, facilitando a inserção de robôs de serviço em um meio social. O sistema de navegação GProxemic consegue obter as coordenadas geográficas bem como as características regionais do local onde o robô está inserido, essas informações são processadas para determinar a zona próxima, realizando a navegação autônoma com o algoritmo social momentum (Mavrogiannis, 2018). Deste modo, com base na localização do robô, é determinada a distância que ele deve respeitar em sua navegação.

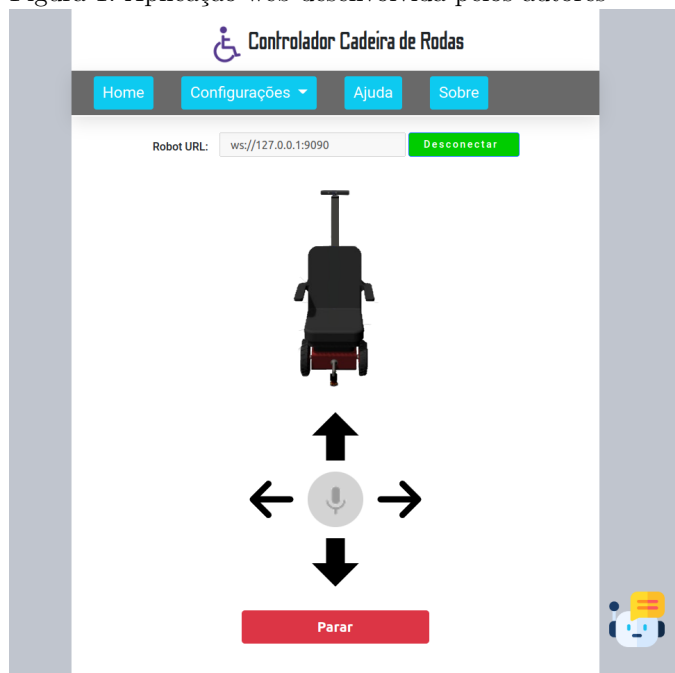
3. ERIKA: ARQUITETURA DE NAVEGAÇÃO COM RECONHECIMENTO SOCIAL BASEADA EM UM CHATBOT

Nesta seção será exposta, a arquitetura por nós desenvolvida, e a implementação do processo de comunicação entre uma Aplicação Web e um Chatbot, utilizado para a interação entre o usuário e um robô de serviço. Para o desenvolvimento do estudo de caso, utilizou-se uma cadeira de rodas motorizada (robô de serviço), simulada no ROS-Gazebo, que irá interagir com o chatbot, por meio da navegação social autônoma.

3.1 Aplicação Web

O objetivo principal da página web desenvolvida, consiste em controlar a simulação da cadeira de rodas no ROS-Gazebo, possibilitando o movimento dessa a partir de comandos de direção através de botões interativos em formato de seta, os quais ao serem pressionados pelo usuário possibilitarão que a cadeira se movimente manualmente na direção selecionada. Isto demonstrado na Figura 1.

Figura 1. Aplicação web desenvolvida pelos autores



Para o desenvolvimento desta utilizou-se como base uma aplicação web localizada no GitHub e desenvolvida pelo Robot Web Tools (Toris, 2015) sendo composta pelas seguintes tecnologias:

— HTML (*HyperText Markup Language*) é uma linguagem de marcação que proporcionou a estrutura do documento da página web.

— CSS (*Cascading Style Sheets*) a qual é uma linguagem de estilo usada para estilizar os elementos que compõem a página.

— JavaScript sendo uma linguagem de programação proporcionou manipular os eventos que ocorrem na página, como, por exemplo, os cliques em botões e integração com a Roslibjs. Esta linguagem ainda possibilitou o uso da API Fetch a qual fornece um método global *fetch*, utilizado para buscar informações advindas da rede de maneira assíncrona, consumindo os dados advindos da API.

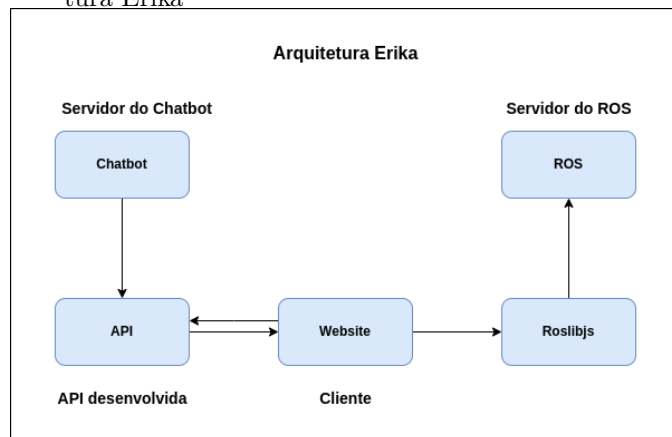
— Bootstrap é um *framework* direcionado ao desenvolvimento *front-end*, usado para a criação dos botões presentes na página.

— Roslibjs (Karaca, 2020) a qual é uma biblioteca JavaScript que permite a interação com ROS a partir do navegador. A mesma recorre ao *WebSockets* para conectar-se ao Rosbridge (Crick, 2017), que fornece uma funcionalidade *JSON API* (Bassett, 2015) auxiliando tanto o ROS, bem como outros programas.

3.2 Arquitetura Erika

A arquitetura Erika desenvolvida é baseada no modelo cliente-servidor, conforme mostrado na Figura 2.

Figura 2. Ilustração do fluxo de comunicação da Arquitetura Erika



O lado do cliente, consiste em uma interface gráfica onde o usuário interage com a aplicação web. Após o cliente encetar com um comando, o chatbot processa a informação e devolve o comando que ele irá executar solicitando ao usuário que selecione o botão verde caso o comando esteja correto para haver a execução da solicitação requisitada. Entretanto, o usuário não confirmando tal dado, ele é solicitado a selecionar o botão vermelho para cancelar.

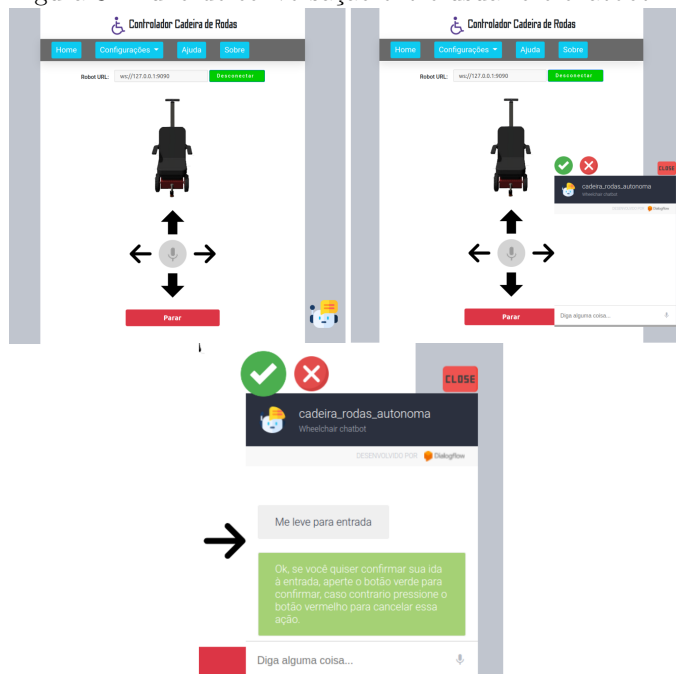
Do lado do servidor, o sistema é subdividido em duas partes, a do chatbot e a do ROS. O servidor do chatbot, interage com uma API conforme ele recebe uma informação inserida pelo usuário, a processa e a manda para essa API. Como contribuição para este artigo, os autores desenvolveram juntamente à arquitetura uma API que foi utilizada para suprir a necessidade dessa interação entre o

Dialogflow e a aplicação *web*. Esta *API* foi elaborada utilizando Node.js, o qual é um *runtime* JavaScript construído no mecanismo JavaScript V8 do Chrome com o *framework* Express.js altamente configurável. Um dos principais motivos da utilização deste *framework* foram seus componentes chamados *middlewares* que permitem a configuração sobre a convenção (Mardan, 2018). Além de proporcionar flexibilidade e alta customização no desenvolvimento de aplicações *web* (Mardan, 2018) e fornecer um conjunto de recursos fundamentais que auxiliaram no desenvolvimento da *API*. Esta *API* é responsável por armazenar o nome da *intent* do Dialogflow correspondente àquele fluxo de conversa atual em um arquivo JSON (*JavaScript Object Notation*).

Para o lado do ROS funcionar é preciso que o usuário confirme a informação, devolvida pelo *chatbot*, selecionando o botão verde contido na página *web*. A partir desta ação, a aplicação *web* faz uma requisição na *API* buscando o nome da *intent* atual que está sendo solicitada no *chatbot* e em seguida, processa e envia para um tópico no ROS através da biblioteca Roslibjs. Ao chegar no ROS, a ação anteriormente escolhida pelo usuário será executada e após a execução a página *web* volta a fazer uma requisição para *API* de modo a limpar o nome da *intent* em uso.

Para que o usuário inicie seu diálogo com o *chatbot*, é preciso acessar a aplicação *web* e em seguida selecionar o ícone do *chatbot* localizado na página. Ao fazer isso, ele pode interagir com o *chatbot* por texto ou voz conforme demonstrado Figura 3.

Figura 3. Fluxo de conversação entre usuário e chatbot



A arquitetura Erika desenvolvida é baseada no modelo cliente-servidor, conforme mostrado na Figura 2.

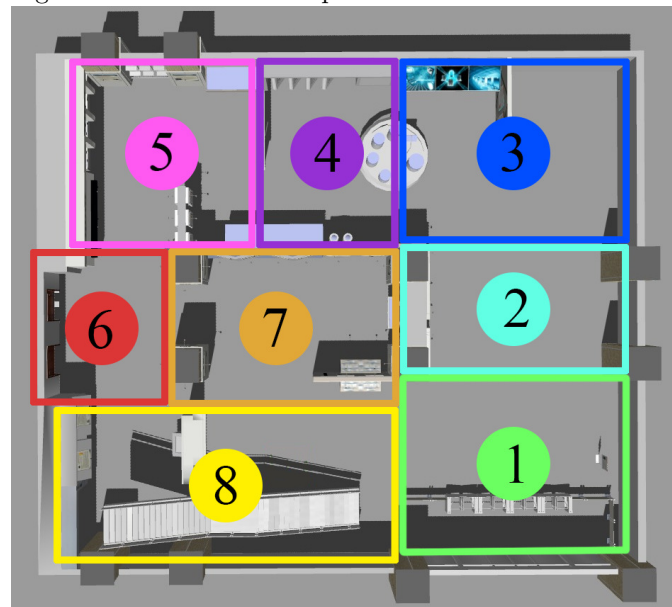
A implementação do fluxo de mensagens do *chatbot* deuse através da ferramenta Dialogflow, uma plataforma do Google que permite a construção de *intents*, ou intenções para o agente conversacional, onde cada *intent* (intenção) permite adicionar frases de treinamento e possíveis

respostas. Neste *chatbot* foram criadas 11 *intents*, das quais 3 são fluxo de conversação básico como mensagem de saudação, despedida e história do museu. As demais *intents* correspondem a locomoção autônoma do robô de serviço no museu no ambiente de simulação ROS-Gazebo com uma navegação proxêmica (Vilasboas, 2021), estas intenções são “Entrada”, “Sala de quadros”, “Bancada tecnológica”, “Itens históricos”, “Banheiros”, “Saída”, “Memorial da humanidade” e “Escada cronológica terrestre”. Além das *intents*, o DialogFlow contém como ferramentas o Agente, o Contexto, o Evento, a Frase de treinamento, Ação e Parâmetros, Resposta, Integrações e o *Fulfillment*. Este último possibilita que a conversa passe entre a solicitação e a resposta do *chatbot* e o usuário (Rosruen, 2018) permitindo a troca de informações com a *API*. A partir destas ferramentas citadas, o uso do *chatbot* torna-se viável e importante, visto que permite uma navegação autônoma, ágil e sem a necessidade repetitiva de cliques nos botões existentes na página.

3.3 Estudo de Caso

De modo a validar a proposta presente no projeto, foi efetuada a implementação da cadeira de rodas autônoma em um museu no ambiente de simulação ROS-Gazebo formado por oito áreas, considerando a entrada como ambiente 1, sala de quadros 2, bancada tecnológica 3, itens históricos 4, banheiros 5, saída 6, memorial da humanidade 7 e por fim a escada cronológica terrestre representada no ambiente 8. Figura 4 apresenta o museu e suas divisões.

Figura 4. Museu e suas respectivas salas



A partir disso constitui-se a primeira hipótese, supondo que o usuário esteja na escada cronológica terrestre e queira se dirigir até a entrada, ele poderá iniciar o fluxo de conversação com o *chatbot*. O usuário inicia o diálogo com o *chatbot* diretamente com um comando para acionar a locomoção do robô, ele entenderá e executará o fluxo contínuo da conversa, sem a necessidade obrigatória de uma mensagem de saudação. Tendo isso em vista, ao informar que gostaria de ir para a entrada, neste instante o *chatbot* retornará “Ok, se você quiser confirmar sua ida à

entrada, aperte o botão verde”. Como se observa na figura 3.

Após a confirmação do usuário, inicializa-se o deslocamento da cadeira de rodas para o ambiente solicitado. Para interagir com a navegação social na cadeira de rodas, utilizamos o sistema Gproxemic (Vilasboas, 2021), respeitando as restrições sociais do ambiente impostas por este sistema. Desse modo, se houver indivíduos no trajeto, a navegação social/proxêmica é aplicada e realiza-se um desvio. Na segunda hipótese, com a chegada à entrada o usuário deseja aproximar-se da placa contendo o mapa e as principais informações do museu, para isso ele usará as setas direcionais contidas na aplicação *web* Figura 1.

Inicialmente o usuário seleciona a seta direcional para a esquerda, de modo a dirigir-se até a placa. Entretanto, percebe a necessidade de aproximar-se um pouco mais para poder visualizar melhor as informações contidas na placa, como, por exemplo, a localização do banheiro. Para isso, ele seleciona a seta direcional para a frente, chegando dessa maneira à posição almejada.

4. DETALHES DE IMPLEMENTAÇÃO E RESULTADOS FUNCIONAIS

Os resultados foram demonstrados em duas diferentes plataformas para comprovar a implementação e eficácia da arquitetura Erika em ambientes de simulação. Utilizou-se o Gazebo/RViz (Vilasboas, 2021) para elucidar a trajetória real realizada pela cadeira de rodas ao redor do mapa e o MATLAB (Vilasboas, 2021) para demonstrar a trajetória da cadeira de rodas a ser feita teoricamente.

Figura 5. Cadeira de rodas na plataforma MATLAB



Fonte: Desenvolvida pelos autores.

Para melhor compreensão, vamos revisar o estudo de caso anteriormente apresentado. O usuário aciona por comando de voz o *chatbot* informando que deseja ir até à entrada do museu, o fluxo da comunicação citado na seção 3.2.1 ocorre e essa informação é publicada em um tópico no ROS como visto na Figura 7. A cadeira faz a navegação da escada cronológica terrestre (área 8) até a entrada (área 1) considerando as restrições sociais conforme é ilustrado na Figura 5 utilizando da Gproxemic que auxilia o deslocamento enviando uma anotação semântica à cadeira de rodas, uma vez que está inserida em um museu onde se encontram pessoas andando pelo ambiente de forma teórica no MATLAB e na Figura 6 pode-se observar este movimento, e de forma prática no Gazebo/RViz.

Figura 6. Cadeira de rodas no Gazebo/RViz



Figura 7. Tópico sendo publicado no ROS com o dado da intent selecionada



As figuras apresentadas anteriormente demonstram que o objetivo da proposta foi alcançado. A arquitetura Erika desenvolvida pelos autores conseguiu estabelecer uma interação entre a cadeira de rodas e o usuário eficientemente, mandando a cadeira para o ponto final da trajetória, podendo ser acionado por comando de voz ou texto. Através do mesmo comprova-se também a funcionalidade e eficiência das setas direcionais presentes na aplicação *web*. Enquanto o Gproxemic manda o ponto geográfico para que o dispositivo reconheça o ambiente onde ele está inserido e possa efetuar a navegação.

5. CONCLUSÃO

Contempla-se a partir dos resultados obtidos através do estudo de caso que a aplicação *web*, a qual contém o *chatbot*, está interagindo de maneira satisfatória com o usuário. A partir disto pode-se inferir que as palavras treinadas estão sendo compreendidas corretamente e as informações estão sendo transmitidas para o ROS-Gazebo permitindo a locomoção da cadeira de rodas no ambiente simulado do museu. Mesmo em situações que hajam pessoas, a cadeira

de rodas consegue respeitar o espaço social das mesmas e fazer desvios que levam o usuário até o destino. Deste modo provou-se que a Arquitetura Erika cumpriu com os objetivos almejados, possibilitando a troca de informações entre o *chatbot*, o usuário, a aplicação *web* e o ROS por meio da API proposta neste artigo.

Em trabalhos futuros pretende-se concretizar este processo através de um Aplicativo Mobile, aumentando desta maneira os mecanismos de acesso ao *chatbot* implementado na aplicação *web*. Além disso, pode-se medir a eficiência da Arquitetura Erika em comparativo com as demais arquiteturas existentes. Desta forma, para validar o algoritmo desenvolvido, o próximo passo é sua implementação em uma cadeira de rodas real.

AGRADECIMENTOS

Agradecemos ao Instituto de Educação, Ciência e Tecnologia da Bahia e ao Grupo de Inovação e Pesquisa em Automação e Robótica - GIPAR por seu apoio e ajuda.

REFERÊNCIAS

- Bassett, L. (2015). *Introduction to JavaScript object notation: a to-the-point guide to JSON*. "O'Reilly Media, Inc."
- Baus, Jörg e Krüger, A.e.W.W. (2002). A resource-adaptive mobile navigation system. In *Proceedings of the 7th international conference on Intelligent user interfaces*, 15–22.
- Crick, Christopher e Jay, G.e.O.S.e.P.B.e.J.O.C. (2017). Rosbridge: Ros for non-ros users. In *Robotics Research*, 493–504. Springer.
- Daza, Marcos e Barrios-Aranibar, D.e.D.A.J.e.C.Y.e.V.J. (2021). An approach of social navigation based on proxemics for crowded environments of humans and robots. *Micromachines*, 12(2), 193.
- Gomes, Bruno Rocha e Jacob Jr, A.F.L.e.P.I.d.J.P.e.C.S. (2020). Ágata: um chatbot para difusão de práticas para educação ambiental. In *Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web*, 85–89. SBC.
- Greenberg, Saul e Marquardt, N.e.B.T.e.D.M.R.e.W.M. (2011). Proxemic interactions: the new ubicomp? *interactions*, 18(1), 42–50.
- Hart, William B e JaLC, D. (2002). Edward t. hall and the history of intercultural communication: the united states and japan sub title author rogers, everett m.(miike, yoshitaka).
- Heryandi, A. (2020). Developing chatbot for academic record monitoring in higher education institution. In *IOP Conference Series: Materials Science and Engineering*, volume 879, 012049. IOP Publishing.
- Karaca, Mustafa e Yayan, U. (2020). Ros based visual programming tool for mobile robot education and applications. *arXiv preprint arXiv:2011.13706*.
- Khaliq, Saran e Ahsan, S.e.N.M.D. (2021). Multi-platform hardware in the loop (hil) simulation for decentralized swarm communication using ros and gazebo. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 310–315. IEEE.
- Kim, Ryan H e Min, H.G. (2021). Development of a conversing, temperature scanning, autonomously navigating robot to help screen for covid-19. In *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, 980–987. IEEE.
- Koubâa, A.e.o. (2017). *Robot Operating System (ROS)*, volume 1. Springer.
- Lalwani, Tarun e Bhalotia, S.e.P.A.e.R.V.e.B.S. (2018). Implementation of a chatbot system using ai and nlp. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST) Volume-6, Issue-3*.
- Lombardi, M e Pascale, F.e.S.D. (2019). An application for cultural heritage using a chatbot. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, 1–5. IEEE.
- Mardan, A. (2018). Using express.js to create node.js web apps. In *Practical Node.js*, 51–87. Springer.
- Mavrogiannis, Christoforos I e Thomason, W.B.e.K.R.A. (2018). Social momentum: A framework for legible navigation in dynamic multi-agent environments. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 361–369.
- Modrzyk, N. (2018). *Building telegram bots: develop bots in 12 programming languages using the telegram bot API*. Apress.
- Ofoeda, Joshua e Boateng, R.e.E.J. (2019). Application programming interface (api) research: A review of the past to inform the future. *International Journal of Enterprise Information Systems (IJEIS)*, 15(3), 76–95.
- Qi, Xiaoguang e Davison, B.D. (2009). Web page classification: Features and algorithms. *ACM computing surveys (CSUR)*, 41(2), 1–31.
- Rosruen, Nudtaporn e Samanchuen, T. (2018). Chatbot utilization for medical consultant system. In *2018 3rd technology innovation management and engineering science international conference (TIMES-iCON)*, 1–5. IEEE.
- Schraft, Rolf Dieter e Schmierer, G. (2000). *Service robots*. CRC Press.
- Setiaji, Hari e Paputungan, I.V. (2018). Design of telegram bots for campus information sharing. In *IOP Conference Series: Materials Science and Engineering*, volume 325, 012005. IOP Publishing.
- Sheridan, T.B. (2016). Human–robot interaction: status and challenges. *Human factors*, 58(4), 525–532.
- Toris, Russell e Kammerl, J.e.L.D.V.e.L.J.e.J.O.C.e.O.S.e.W.M.e.C.S. (2015). Robot web tools: Efficient messaging for cloud robotics. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 4530–4537. IEEE.
- Vilasboas, João Pedro e Sampaio, M.S.C.e.M.G.F.e.S.A.B.e.D.A.J.a.B.A.D.e.C.Y.e.S.J.E. (2021). Application of social constraints for dynamic navigation considering semantic annotations on geo-referenced maps. In *IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society*, 1–7. IEEE.
- Wang, H e Lei, Z.e.Z.X.e.Z.B.e.P.J. (2016). Machine learning basics. *Deep learning*, 98–164.
- Yusoff, Mohd Najwadi e Dehghantanha, A.e.M.R. (2017). Forensic investigation of social media and instant messaging services in firefox os: Facebook, twitter, google+,

telegram, openwapp, and line as case studies. In *Contemporary digital forensic investigations of cloud and mobile applications*, 41–62. Elsevier.