

Doutora Sara: um assistente virtual para marcação de consultas médicas

André O. Silva¹, Leonardo B. Campos²

¹Engenharia Elétrica – Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) – Campus Vitória da Conquista – Vitória da Conquista – BA – Brasil

²Sistemas de Informação – Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) – Campus Vitória da Conquista – Vitória da Conquista – BA – Brasil

{badecoandre1999, leonardobcampos}@gmail.com

Abstract. *The wide insertion of applications for instant messaging exchanges has enabled the spread of virtual assistants and chatbots. In the health area, an important application of these robots occurs in scheduling medical appointments where, commonly, there is a need for a person to perform this task. Thus, Dr. Sara is a virtual assistant for scheduling medical appointments through the WhatsApp APP, capable of processing text in natural language. The virtual assistant uses machine learning and can be easily integrated with home automation devices, such as Alexa, and interact with the patient through natural language.*

Resumo. *A ampla inserção dos aplicativos para trocas instantâneas de mensagens têm viabilizado a disseminação de assistente virtuais e chatbots. Na área de saúde, uma importante aplicação desses robôs ocorre no agendamento de consultas médicas onde, comumente, há necessidade de uma pessoa para executar essa tarefa. Dessa forma, Doutora Sara é um assistente virtual para agendamento de consultas médicas por meio do aplicativo WhatsApp, capaz de processar texto em linguagem natural. O assistente virtual utiliza aprendizagem de máquina e pode ser facilmente integrado a dispositivos de automação residencial, como o Alexa, e interagir com o paciente por meio de linguagem natural.*

1. Introdução

O amadurecimento de novos recursos tecnológicos, como a Inteligência Artificial (IA), vem favorecendo a criação de sistemas cada vez mais autônomos, dentre eles, destacam-se os assistentes virtuais, ou *chatbots*. Os assistentes virtuais são programas de computador, ou softwares, que permitem automatizar interações entre seres humanos e máquinas (Cunha, 2016). Estas interações podem acontecer de diferentes formas, tais como: voz, escrita em linguagem natural, sensores de movimento, entre outros.

Com os avanços dos aplicativos de mensagens, da IA e a necessidade de personalização extrema, está sendo criado um cenário favorável para a popularização dos assistentes virtuais, sendo capazes de tornarem-se substitutos dos aplicativos atuais (Sondergaard, 2016). Segundo Insider (2016), a quantidade de usuários ativos nos aplicativos mensageiros ultrapassou a quantidade de usuários ativos nas redes sociais a partir de 2015.

Devido ao uso massivo destes aplicativos de mensagens, inúmeras empresas estão vislumbrando um grande potencial para lançar novos serviços diferenciados nestes canais. Em virtude do uso crescente e contínuo destas plataformas, os aplicativos de mensagens estão, naturalmente, transformando-se em uma espécie de “central” para grande parte de serviços. Um domínio onde os assistentes virtuais têm-se mostrado apropriados é o da saúde (Buinhas et al., 2018), (Balsa et al., 2019), (Carvalho et al., 2019).

Nesse contexto, este artigo apresentará um assistente virtual no domínio da saúde capaz de prover o primeiro nível de atendimento ao cliente, ou seja, realizar a marcação de consultas médicas sem necessidade de intervenção humana e interpretando as instruções contidas em frases enviadas em linguagem natural. Além disso, o assistente virtual é capaz de interagir com os pacientes por meio da voz utilizando os assistentes residenciais, tais como Google Home e Alexa da Amazon.

2. Doutora Sara

Doutora Sara é um assistente virtual para marcação de consultas médicas desenvolvido para interagir com diferentes aplicativos de trocas de mensagens instantâneas, sobretudo, com o WhatsApp, o aplicativo mais popular entre os brasileiros (Mobile Time, 2020). Por meio do WhatsApp o paciente pode consultar, em linguagem natural, as especialidades médicas disponíveis para marcação da consulta. Em seguida, os dados passados ao mensageiro são processados pelo DialogFlow que envia os dados com a confirmação de marcação da consulta médica ao Banco de dados, conforme mostra a Figura 1.

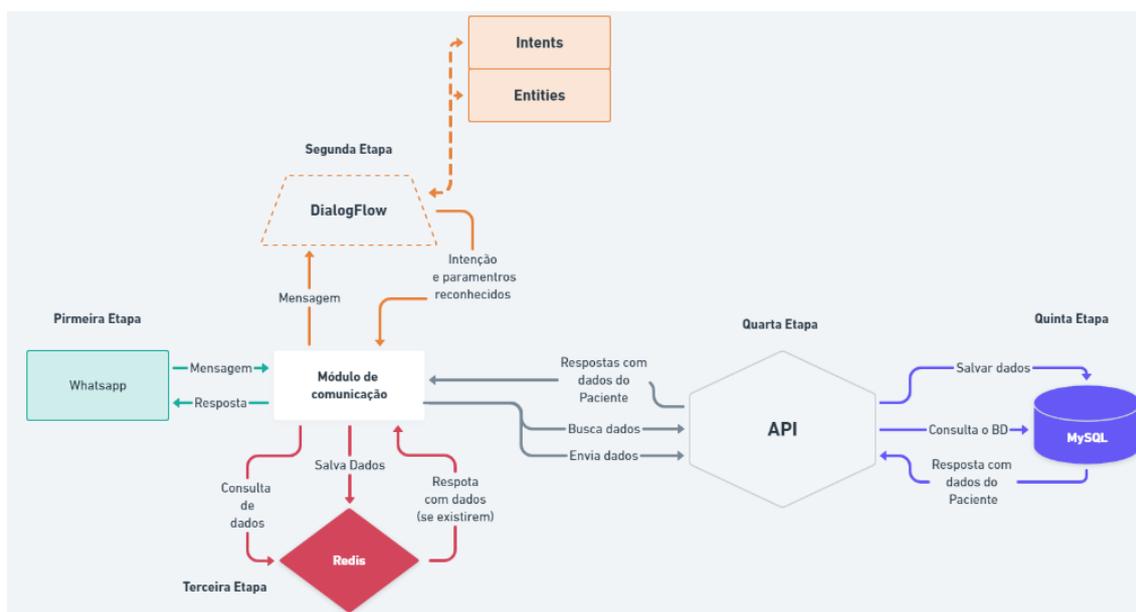


Figura 1. Fluxo de dados do módulo de comunicação

2.1 Primeira Etapa: WhatsApp

O WhatsApp é o maior aplicativo de comunicação do mundo e que durante a pandemia do novo Coronavírus, em janeiro de 2020, conseguiu a marca inédita de estar presente em 99% dos smartphones brasileiros (Jucá e Lopes, 2020).

O WhatsApp está pavimentando o caminho para se tornar um super-app. Primeiro, em agosto de 2018, abriu sua API para que grandes empresas pudessem utilizá-lo como

canal na comunicação com os consumidores. E mais recentemente, o Banco Central do Brasil autorizou a realização de transferências bancárias por meio do WhatsApp (Calil, 2021).

2.2 Segunda Etapa: DialogFlow

O Dialogflow é uma plataforma de compreensão de linguagem natural usada para projetar e integrar uma interface de usuário conversacional em aplicativos para dispositivos móveis, sistemas web, *chatbots* e até sistemas de respostas por voz interativos, como os assistentes residenciais Google Home e Alexa.

Essa plataforma usa aprendizado de máquina (*machine learning*) e processamento de linguagem natural (*Natural Language Processing* – NLP) para interpretar o texto em linguagem natural fornecido pelo usuário e, a partir dele, identificar tanto a intenção do usuário, quanto extrair alguns parâmetros do texto fornecido e identificar ações através do contexto de conversa. O DialogFlow tem quatro conceitos básicos, esses são:

- *Intents*: Uma *intent* categoriza a intenção de um usuário. Para cada agente, é necessário definir várias *intents*, em que suas *intents* combinadas podem lidar com uma conversa completa. Quando um usuário final escreve ou diz algo, o Dialogflow corresponde o conteúdo à melhor *intent* do seu agente;
- *Entities*: Cada parâmetro de *intent* tem um tipo, chamado tipo de entidade, que dita exatamente como os dados de uma expressão de usuário final são extraídos. O Dialogflow fornece entidades predefinidas do sistema que podem corresponder a muitos tipos comuns de dados. Por exemplo, há entidades do sistema que correspondem a datas, horários, cores, endereços de e-mail e assim por diante. Também é possível criar suas próprias entidades personalizadas para corresponder a dados personalizados. Por exemplo, nós definimos uma entidade chamada *plans*, que define todos os planos registrados no sistema. Assim, o agente consegue reconhecer os planos registrados quando o usuário o informar e responder como um parâmetro da intenção;
- *Contexts*: Os contextos do Dialogflow são semelhantes ao contexto da linguagem natural. Se alguém diz "eles são azuis", você precisa de contexto para entender a que "eles" está se referindo. Da mesma forma, para o Dialogflow processar uma expressão de usuário final como essa, ele precisa de contexto para corresponder corretamente a uma *intent*;
- *Fulfillment*: Por padrão, o agente do DialogFlow responde a um *intent* correspondente com uma resposta estática, mas é possível modificar isso habilitando o *Fulfillment*. O *Fulfillment* é a parte de código do DialogFlow, dessa maneira, quando for habilitado, o Dialogflow envia uma requisição HTTP do tipo POST para um Webhook, que concentra toda a lógica do seu *chatbot*. A partir daí é possível realizar consultas ao BD, requisições à APIs externas e afins.

Ao criar um agente conversacional, é necessário inserir várias frases de treinamento para cada *intent* criado. As frases de treinamento servem como um ponto de partida para o agente. Daí em diante a IA, usando *machine learning* e o NLP, vai tentar identificar as intenções dos usuários baseado nas frases informadas no treinamento do agente.

Quando um usuário consulta o agente, o *chatbot* tenta combinar as frases definidas no treinamento para corresponder à frase do usuário. Com base nessas frases, o *chatbot*

responderá às solicitações do usuário. Se as frases de treinamento são descritas de várias maneiras, o *chatbot* pode responder de forma mais eficaz.

2.3 Terceira Etapa: Módulo de Comunicação

A implementação do módulo de comunicação foi essencial para o funcionamento do Sistema, pois o Whatsapp não tem uma API pública de comunicação gratuita. A API fornecida pelo Whatsapp é destinada apenas para grandes empresas e essa verificação ainda é feita manualmente pelo setor responsável do aplicativo.

Os intermediários são provedores oficiais do whatsapp, empresas parceiras da plataforma que conseguem disponibilizar *chatbots* e até mesmo uma API completa. Mas os valores cobrados por esses serviços giram em torno de R\$2.000,00 por mês. Como o Whatsapp web utiliza de certificados TSL, o servidor não aceita requisições que não sejam feitas pelo endereço <https://web.whatsapp.com>, impossibilitando, assim, o envio de requisições por terceiros. A solução mais viável para essa dificuldade foi criar um cliente do Whatsapp web, usando uma biblioteca que automatiza as ações de um navegador. Daí chegamos na biblioteca Puppeteer, para NodeJs.

O módulo usa a biblioteca Puppeteer, que na prática trata-se de uma biblioteca NodeJs, fornece uma API de alto nível para controle programático do navegador Chromium por meio do protocolo DevTools. Assim, todas as ações que podem ser feitas através de um navegador, podem ser feitas usando o Puppeteer, mas de forma automatizada. O Puppeteer pode ser utilizado para fazer Web Scraping, que é o ato de baixar automaticamente os dados de uma ou mais páginas web, tendo como principal objetivo extrair informações muito específicas. Também é possível automatizar ações realizadas via uso da interface do navegador, tais como preencher e submeter formulários ou até mesmo simular a navegação de um usuário dentro do site.

O módulo de comunicação utiliza de um banco de dados em memória, Redis, para controle de sessão. Desta forma, assim que o usuário inicia a conversa com o *chatbot*, o módulo de comunicação busca na API os dados do usuário, caso exista. Caso o usuário já tenha marcado alguma consulta com a Doutora Sara, os dados dele, como nome e plano de saúde, já estarão salvos. A partir daí a comunicação passa a ser personalizada para ele.

Todos os dados são salvos no Redis com um tempo de expiração de 20 min, este controle de sessão é feito por alguns motivos. O primeiro motivo pelo qual os dados salvos no Redis são temporários é porque o Redis é um BD em memória, desta forma os dados preenchidos lá são muito caros para aplicação. A memória é muito volátil e é manipulada constantemente pela aplicação, além de ser um recurso mais escasso do que salvar os dados em disco. Por isso os dados salvos em memória têm de ser temporários, para que os dados de outros usuários possam usar aquele espaço de memória em sequência.

O segundo motivo para o tempo de expiração de sessão ser de 20 minutos, é porque os contextos de conversa do Dialogflow expiram em 20 minutos por *default*. Então, não faz sentido para a aplicação os dados continuarem salvos fora do contexto da conversa. A principal vantagem de usar um BD em memória é a velocidade de consulta. Buscar dados salvos em memória é muito mais rápido que buscar os dados em disco ou até mesmo de uma API. Por isso, a consulta à API é feita no início da conversa e as consultas subsequentes são feitas diretas à memória da aplicação, deixando as requisições

à API apenas para casos excepcionais, se esses valores não existirem em memória, resultando, assim, em uma comunicação mais rápida com o usuário.

2.4 Quarta Etapa: API de Comunicação com o WhatsApp

Para viabilizar a comunicação do assistente virtual com o aplicativo WhatsApp foi necessário o desenvolvimento de uma API (*Application Programming Interface*). A criação da API fez-se necessária, pois facilita o desenvolvimento do projeto para que este tenha um único *backend* que se comunica tanto com o *chatbot*, quanto com a aplicação web. A API foi desenvolvida utilizando a linguagem de programação JavaScript e o NodeJs.

A API abstrai para si todas as regras de negócio do sistema e disponibiliza através de um servidor HTTP uma série de rotas de comunicação. Por meio das requisições dos tipos PUSH, DEL, PUT, GET é possível acessar e modificar os dados do BD. O acesso à API é protegido por um Json Web Token. Este *token* permite o acesso a algumas rotas do sistema que são utilizadas pela aplicação web. Ao fazer o login, o usuário (clínica) recebe o *token* que é vinculado a todas suas requisições à API a partir daí. No *backend*, o *token* é validado a cada requisição recebida e tem duração de 12 horas.

2.5 Quinta Etapa: Banco de Dados

Por fim, foram criadas dez tabelas em um Banco de Dados relacional para que o sistema armazenasse os dados referentes aos relacionamentos existentes entre as tabelas especificadas.

3. Resultados e Discussão

Essa seção apresenta uma rotina para marcação de uma consulta médica. A Figura 2 mostra o início de uma conversa para marcação de uma consulta com o especialista em urologia.

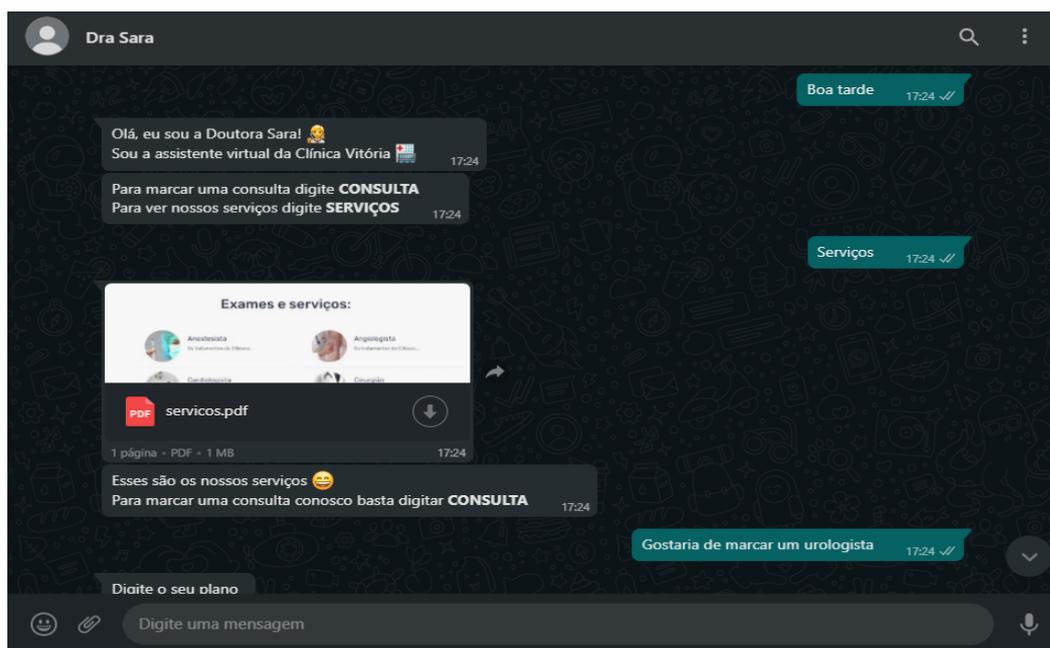


Figura 2. Interações iniciais com a assistente virtual para marcação de uma consulta médica na especialidade de urologia

Em seguida, são apresentadas informações sobre os possíveis planos de saúde ao qual o paciente seja beneficiário, conforme mostra a Figura 3.

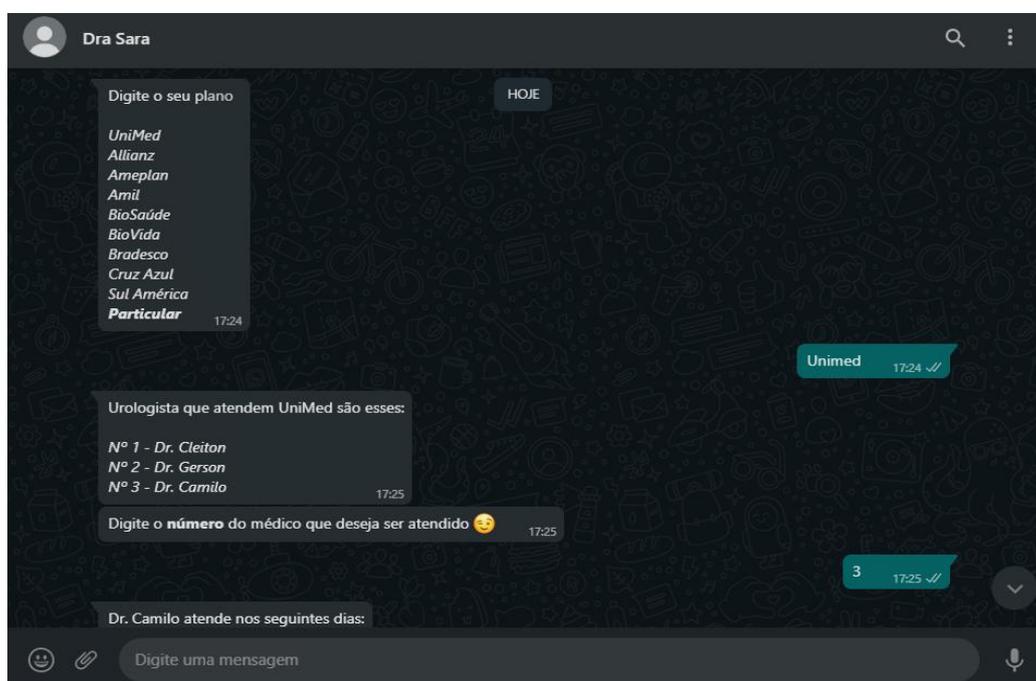


Figura 3. Interações com a assistente virtual para escolha dos médicos filiados aos planos de saúde disponíveis

Usando contextos, é possível controlar o fluxo de uma conversa. Para configurar contextos para uma *intent*, é necessário configurar contextos de entrada e saída. Quando uma *intent* é correspondida, todos os contextos de saída configurados para essa *intent* são ativados. Embora os contextos estejam ativos, é mais provável que o Dialogflow corresponda às *intents* configuradas com contextos de entrada correspondentes aos contextos ativos no momento.

Por exemplo, se você definir um número como referência a algum valor, como número 1 corresponder ao médico “João” e o número 2 corresponder ao médico “Paulo”. Caso você tenha algum outro parâmetro que também é correspondido por um número, como número 1 corresponder ao plano “Unimed” e o número 2 corresponder ao plano “Allianz”, o input “1” do usuário pode ser reconhecido tanto como “João”, quanto como “Unimed”, a depender do contexto ativo na conversa.

Por fim, a Figura 4 mostra a etapa final do processo de marcação da consulta com a apresentação dos dias e horários disponíveis e o preenchimento dos dados pessoais do paciente. O Dialogflow foi usado como uma biblioteca de processamento de linguagem natural, criando um agente conversacional na plataforma e se comunicando com o agente através de requisições HTTP. O agente denominado Doutora Sara responde com os parâmetros extraídos da conversa e com a intenção que ele imagina que o usuário quis expressar baseado em uma série de intents criadas na plataforma, como **(i)** Marcar uma consulta; **(ii)** Informar o médico escolhido; **(iii)** Informar a data selecionada, e etc.

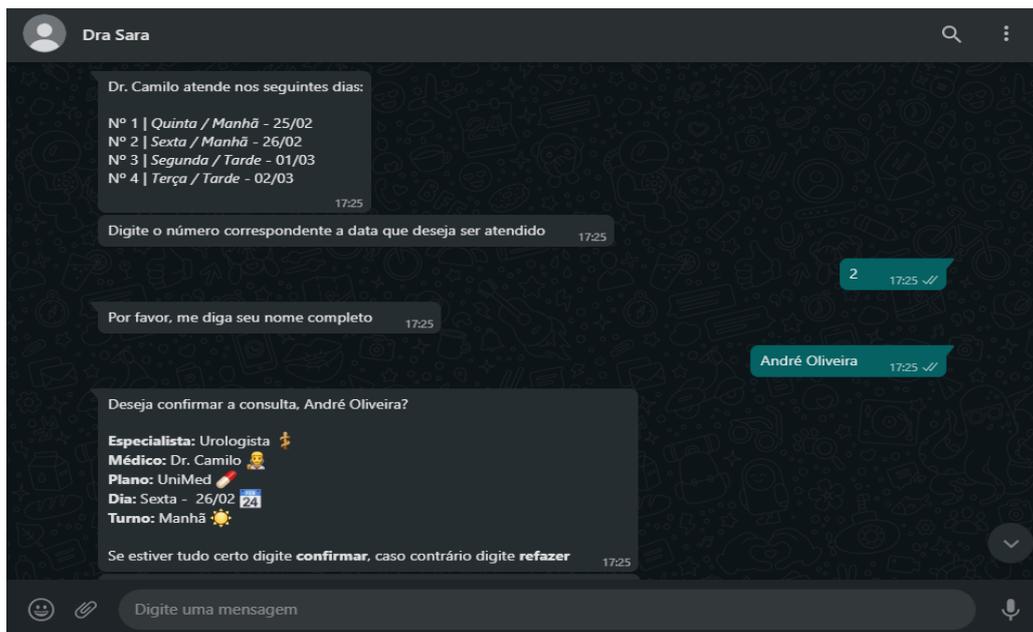


Figura 4. Interações com a assistente virtual para escolha do dia e horários disponíveis, bem como, preenchimento de dados pessoais do paciente

Por fim, o Dialogflow permite a integração com o Google Assistant, assim, é possível acionar o *chatbot* através do assistente residencial Google Home ou até mesmo smartphones e televisores inteligentes com o sistema operacional Android. Usando a integração com o Google Assistant é possível se comunicar com a Doutora Sara por voz, conversando naturalmente como em uma conversa real. A Doutora Sara também responde por voz, tornando assim a conversa muito mais fluida.

O fluxo da conversa no Google Assistant é um pouco diferente por conta da limitação de um sistema por voz. Não é possível enviar um arquivo ou foto listando os médicos ou especialistas disponíveis para o atendimento, assim como também fica confuso solicitar números de referência para o usuário. As informações têm que ser solicitadas todas separadamente e uma de cada vez, para não confundir o usuário final.

Por exemplo, não dá para solicitar o dia e o turno em que o usuário deseja ser atendido de uma só vez, assim como é feito pelo Whatsapp, o usuário iria acabar se confundindo, pois não existe uma interface visual para que o mesmo recorra. Neste ambiente a data e o turno da consulta são solicitados separadamente. Mas, no final, as requisições para a API e as consultas ao banco são basicamente as mesmas.

Não há necessidade de um BD em memória com o Google Assistant pois não existe modo de espera (*standby*), assim que o usuário fica um tempo sem responder o próprio Google Assistant encerra o atendimento. Ao contrário do módulo de comunicação, que utiliza o DialogFlow apenas como uma biblioteca de NLP, o *chatbot* por voz usa a integração nativa do DialogFlow para interagir com o Google Assistant, portanto, fez-se necessário utilizar o recurso de código do DialogFlow, o Fulfillment, para personalizar as respostas enviadas para o usuário final. Habilitando o Fulfillment é possível realizar a comunicação necessária com a API.

4. Conclusão

O presente trabalho apresentou um assistente virtual intuitivo e inteligente para marcação de consultas médicas. Doutora Sara pode ser usado em clínicas, hospitais ou consultórios e pode ser facilmente integrado a dispositivos de automação residencial, como o Google Home ou o Alexa da Amazon, e interagir com o paciente por meio de linguagem natural.

Como trabalhos futuros é possível integrar Doutora Sara a outros assistentes virtuais em um contexto de Cidades Inteligentes, onde um grande emaranhado de serviços e informações podem ser prestados ao cidadão em um único número de telefone. Uma assistente virtual de tudo e para todos.

Referências

- Balsa, J. et al., "Intelligent Virtual Assistant for Promoting Behaviour Change in Older People with T2D." EPIA Conference on Artificial Intelligence. Springer, Cham, 2019.
- Buinhas, S. et al., Virtual assistant to improve self-care of older people with type 2 diabetes: First prototype. International Workshop on Gerontechnology. Springer, Cham, 2018.
- Calil, J., Banco Central autoriza transferências bancárias pelo WhatsApp. 2021. Disponível em: <<https://g1.globo.com/economia/tecnologia/noticia/2021/03/30/banco-central-autoriza-pagamentos-e-transferencias-pelo-whatsapp.ghtml>>
- Carvalho, L. et al., Dr.^a Lara: Assistente Virtual de Apoio e Acompanhamento ao Pré-Natal, Conferências IADIS Ibero-Americanas WWW/Internet e Computação Aplicada 2019, 2019.
- Cunha, D., Guia definitivo para criar o seu Chatbot cognitivo sem programação. 1. ed. São Paulo: [s.n.], 2016.
- Insider, B., Messaging apps are now bigger than social networks. 2016. Disponível em: <<http://tableless.com.br/o-que-nodejs-primeiros-passos-com-node-js>>.
- Jucá J., Lopes L., Estudo aponta que Whatsapp é o aplicativo mais usado durante a pandemia. 2020. Disponível em <<https://www.cnnbrasil.com.br/tecnologia/2020/08/03/estudo-aponta-que-whatsapp-e-o-aplicativo-mais-usado-durante-a-pandemia>>
- Mobile Time, Uso de APPs no Brasil, Dezembro de 2020. 2020. Disponível em: <<https://www.mobiletime.com.br/pesquisas/uso-de-apps-no-brasil-dezembro-de-2020/>>
- Sondergaard, P., Para Gartner, não haverá mais aplicativos em 2020. 2016. Disponível em: <<http://ofuturodascoisas.com/gartner-em-2020-nao-havera-mais-aplicativos>>.