

IMPLEMENTAÇÃO DE UM CHATBOT UTILIZANDO A PLATAFORMA RASPBERRY PI

Jean Victor Mendes dos Santos¹, Rogério Ferreira Sgoti², Marco Antonio Nagao³

¹Tecg.º em Análise e desenvolvimento de sistemas, FATEC Botucatu, jean-w220@hotmail.com

²Docente da Faculdade de Tecnologia de Botucatu, FATEC, rsgoti@fatecbt.edu.br

³Docente da Faculdade de Tecnologia de Botucatu, FATEC, mnagao@fatecbt.edu.br

RESUMO

Com o avanço da tecnologia, está cada vez mais comum o uso de comandos de voz para executar funções rotineiras. Deste modo, este trabalho teve como objetivo o desenvolvimento de um *chatbot* que reconhece voz, mantém uma conversação e executa funções programadas pelo usuário. Como ferramenta principal do *software*, utilizou-se o *framework VictroBrain*, desenvolvido na linguagem de alto nível PHP (*Hypertext Preprocessor*), com conexão a um banco de dados MySQL utilizando a interface PDO (*PHP Data Object*), tecnologias de detecção de tom da *IBM Watson* e reconhecimento de voz no idioma inglês através da *API (Application Programming Interface) Web Speech*, através do navegador. Por meio destas ferramentas e tecnologias obteve-se um *software chatbot* denominado *VictroRobot* cujo principal objetivo é o reconhecimento de voz e alteração de humor. Como metodologia de desenvolvimento foi utilizada a prototipagem. Mediante o desenvolvimento e testes deste software (*chatbot*), concluiu-se que o objetivo do trabalho foi atingido. O uso do *framework VictroBrain* possibilitou uma codificação padronizada e facilitada, pois conta com funções básicas prontas, demonstrando ser uma ferramenta útil para desenvolvedores.

Palavras-chave: Chatbot. VictroBrain. VictroRobot.

ABSTRACT

IMPLEMENTATION OF A CHATBOT USING RASPBERRY PI PLATFORM

With the technology advancement, it is increasingly common to use voice commands to perform routine functions. In this way, this work had as objective the development of a chatbot which recognizes voice, keeps a conversation and runs user programmed functions. As main software tool, the VictroBrain framework, developed in the high level language PHP (Hypertext Preprocessor), with connection to a MySQL using the interface PDO (PHP DataObject), tone detection technologies from IBM Watson and voice recognition in the English language through the Application Programming Interface (API) Speech through the web browser. By means of these tools and Technologies produced a chatbot software named VictroRobot whose main objective is the recognition of voice and mood change. Through the development and testing the software, it was concluded that the objective of this work was completed. The use of the framework VictroBrain allowed an easy standard coding, it counts with basic functions, showing to be a useful tool for developers.

Keywords: Chatbot. VictroBrain. VictroRobot.

1 INTRODUÇÃO

Há tempos, na história humana, a criação de robôs é uma busca constante. Com a evolução das tecnologias digitais de informação e comunicação surge a possibilidade de desenvolver um *chat* – software de conversa online – que seja robotizado, ou seja, um *chatbot* (*chat* + *robot*). De modo mais técnico, um *chatbot* é um software desenvolvido para simular uma conversação inteligente com um ou mais usuários. Esta conversa pode ocorrer por meio de texto ou áudio e os *chatbots* podem ser programados para manter tanto conversas básicas – como as que ocorrem no cotidiano de uma pessoa – quanto responder questões específicas sobre algum tema pré-programado.

Nesse sentido, esse tipo de sistema compreende o contexto do que foi dito ou escrito, e entrega uma resposta baseada na mensagem entendida. Segundo Gupta et al. (2015), um *chatbot* é um exemplo de aplicação de inteligência artificial, uma vez que os mesmos podem aprender outras formas de entender o contexto e assimilar respostas dadas anteriormente. Para Ferreira e Uchôa (2008), a inteligência artificial é uma área de conhecimento humano que busca uma forma de reproduzir a mente humana utilizando elementos computacionais. Como detalhar a inteligência artificial não faz parte do escopo deste trabalho, para mais definições e aplicações sobre essa área, recomenda-se uma consulta a obra de Sellitto (2002).

Por meio da inteligência embutida no software objeto deste trabalho, atividades rotineiras residenciais ou comerciais podem ser auxiliadas pelo referido sistema. Assim, usuários interessados poderão contar com um *chatbot* programável que pode auxiliar a gerenciar algumas de suas tarefas cotidianas, interagir com outros equipamentos e facilitar toda a organização de uma casa e/ou de uma empresa. O *chatbot* apresentado e discutido no presente resumo é resultado do trabalho de conclusão de curso do primeiro autor para obtenção do título de tecnólogo, em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia de Botucatu. Após averiguar que a capacidade de reconhecimento de voz é um recurso cada vez mais adotado quando do desenvolvimento de aplicativos móveis, surgiu a motivação para esta pesquisa. Apurou-se também que este recurso conta com aprimoramento constante e o seu custo está cada vez menor, possibilitando a um maior número de usuários o seu acesso (RATO, 2016).

O objetivo do presente trabalho foi o desenvolvimento de um *chatbot* inteligente denominado “Victro Robot”, com utilização de hardware de baixo custo e que, aplicado a processos de conversação com usuários, apresenta comportamento similar com o comportamento humano.

2 MATERIAL E MÉTODOS

O hardware que dá suporte ao *chatbot* desenvolvido é o *Raspberry Pi*. Ele é um *hardware* de baixo custo desenvolvido pela Fundação *Raspberry Pi* e sua primeira versão foi lançada em 10 de fevereiro de 2012. O principal propósito deste hardware é promover o ensino de Ciências da Computação em escolas e faculdades em países desenvolvidos. Entretanto, esse tipo de tecnologia está se tornando conhecida no Brasil.

Segundo Simioni e Bertini (2014), o *Raspberry Pi* é um microcomputador capaz de fazer processamento de voz com 1GB de memória RAM, além de ter a capacidade de gerenciar componentes e sensores através de GPIO (*General Purpose Input/Output*) e USB (*Universal Serial Bus*). Desse modo, ele pode ser programada e suportar aplicativos que podem ser desenvolvidos em diversas linguagens como *Python*, PHP, *Java*, entre outras (SIMIONI; BETINI, 2014).

Para o levantamento de requisitos foi feito um levantamento e testes de outros *chatbots* e assistentes pessoais, como os assistentes da empresa *Google* (*Google Now* e *Google Assistant*) e da *Apple* (*Siri*).

Para o desenvolvimento do *chatbot VictroRobot* foi utilizada a metodologia de prototipagem. No hardware *Raspberry Pi* foi instalado o LAMP – sigla para *Linux*, *Apache*, *MySQL* e *PHP* – para proceder com a instalação do *framework VictroBrain*, obtendo-se, então, o funcionamento do protótipo do referido *chatbot*.

VictroBrain é o *software* que comanda o *chatbot*. Ele é um *framework* PHP desenvolvido especialmente para o *chatbot VictroRobot*. Para Minetto (2007), um *framework* é um *software* “base” com o qual pode ser desenvolvido outros *softwares*. Esses outros *softwares* utilizam funções, técnicas e metodologias do *framework* e, com isso, o desenvolvimento se torna mais rápido e segue um padrão (MINETTO, 2007).

3 RESULTADOS E DISCUSSÕES

3.1 Instalação do *framework*

O processo de instalação do *framework* é simples, não sendo necessária nenhuma programação e todo o processo é visual e interativo ao usuário. É preciso acessar o *site* oficial do *framework*: victrobrain.com e clicar no botão *download* para obter a última versão do sistema. Será copiado um arquivo *tar.gz* com o sistema, sendo necessário um servidor *Linux* com *Apache 2.2*, o *PHP 7.1* ou posterior e um sistema gerenciador de banco de dados (SGBD) compatível com o sistema.

3.2 VictroBrain

A tela inicial do *framework* é dividida em 4 partes: menu lateral esquerdo, menu superior, menu lateral direito (oculto) e o carregador de conteúdo. Na tela inicial é possível visualizar os componentes básicos do sistema. Este *design* pode ser carregado ou não, ou seja, caso o programador queira desenvolver outro *design* para sua aplicação o *design* padrão não será exibido.

Os comandos do *framework* são executados através de um terminal próprio do sistema onde é possível alterar informações do sistema como título, local de instalação, usuário padrão, entre outras informações e também instalar *plug-ins*, *helpers* e *add-ons*.

3.3 Plug-ins (Robot)

Os *plug-ins* são a principal parte do sistema, eles são chamados de *robot* no sistema. O *robot* é o produto que se deseja desenvolver, ou seja, o *software* em si. No *VictroRobot* foi criado um *robot* para detecção de fala, pois esta é uma grande função; para pequenas funções é utilizado os *helpers* que são chamados de *power* no sistema.

Para se desenvolver um *robot* no sistema é necessário que algumas regras/padrões de desenvolvimento sejam seguidas. Primeiramente é necessário criar a pasta do *robot* dentro da pasta *victro_apps/victro_robot*. Dentro da pasta criada para o *robot* é necessário mais duas pastas, uma chamada *model*, onde são armazenados os arquivos que lidam com o banco de dados, sessões e outras funções, e outra chamada *view* que armazenará arquivos de visualização para o usuário.

3.4 Helper (Power)

O *helper* é chamado de *power* no sistema e sua principal função é o uso de pequenas funções para agregar nos *robots* (*plug-ins*). *Powers* não podem inserir dados no banco de dados, mas podem ter sua própria tabela e fazer consultas para os *robots*. O desenvolvedor deverá informar que o *robot* necessita de um *power* para funcionar adequadamente. Para a criação de um *power* basta criar uma pasta dentro de *victro_apps/victro_power* com um arquivo PHP chamado *function.php*.

3.5 Add-ons

Add-ons são *hardwares* externos que podem interagir com o *framework* através de redes *wi-fi*. O hardware externo precisa estar utilizando um código fornecido pelo

próprio *framework* para que esta interação aconteça. O módulo aconselhado pelo *framework* é o NodeMCU ESP8266 e o *framework* disponibiliza o código para ser executado no módulo e deixá-lo compatível com o sistema. Os *add-ons* são controlados através de um *robot (plugin)*. A junção de *add-ons* e *robot* pode resultar em uma automação residencial, por exemplo.

3.6 Softwares existentes de reconhecimento de voz

Segue abaixo uma breve descrição dos principais softwares existentes no mercado para aplicações de reconhecimento de voz.

- **Siri (Apple):** A integrante mais antiga quando se trata de assistente pessoal com reconhecimento de voz, tem seu aprimoramento a cada nova versão do sistema IOS. Esse mecanismo atualmente pode receber comandos de voz tanto para funções básicas do sistema como enviar um SMS como também pode executar funções em aplicativos de terceiros como executar uma música no *Spotify* ou enviar uma mensagem pelo *WhatsApp*. A Siri pode ser executada tanto no iPhone quando no iPad, iWatch e computadores Apple.

- **Cortana (Microsoft):** Lançada junto com o *Windows 8.1* a assistente pessoal da Microsoft tem grande expectativa do mercado. O ponto forte da Cortana é sua capacidade de falar frases inteiras com uma naturalidade. Suas principais funções são chamadas telefônicas, previsão do tempo e alarmes.

- **Google Now:** O assistente pessoal do *Google* é o assistente pessoal mais utilizado no momento por ser o assistente pessoal do sistema operacional móvel mais utilizado, o *Android*. A grande vantagem desse assistente é a naturalidade dele para reconhecer o idioma português. Ele pode executar funções básicas do sistema *Android* e abrir aplicativos instalados.

- **Google Assistente:** O novo assistente do *Google*, disponível para os SO *Android* versão 6 e posteriores. Ele traz todas as facilidades do *Google Now*, mas agora trabalhando com inteligência artificial.

Google Home: É destinado para ficar parado em um cômodo da residência. Ele é capaz de interagir via comandos de voz com as pessoas, mantendo uma conversação e gravando preferências por pessoa, além de interagir com outros equipamentos na casa como *Smart TV*.

Alexa (Amazon): Alexa é a assistente pessoal virtual com mais “habilidades” a oferecer ao usuário. São mais de 15 mil funções de interação com outros aplicativos.

No *Victro Robot* a chamada para reconhecimento de voz acontece no seu processo de inicialização. Assim que o *Raspberry Pi* é iniciado é feita uma requisição para iniciar o navegador *Chromium* (2017), e, logo após, é chamado o *framework VictroBrain* que inicia uma página HTML5, chamando o script e, assim, começando o reconhecimento de voz. Todo esse processo de inicialização do *Raspberry Pi* e navegador com *framework*, demora cerca de 60 segundos. Segundo o site oficial do *Chromium* (2017), este navegador é um projeto de código aberto que visa construir uma maneira mais segura, rápida e estável para todos os usuários de Internet experimentar a web. Trata-se de uma versão *Open-Source* do navegador *Google Chrome*.

3.7 IBM Watson Tone Analyser

Para integrar a API ao *framework* foi desenvolvido um *robot (plugin)*, responsável por fazer a chamada do serviço via cURL (libcurl) do PHP. De acordo com a documentação oficial, o PHP 2017 suporta libcurl, uma biblioteca criada por Daniel Stenberg, que permite a conectividade e a comunicação com diferentes tipos de servidores usando diferentes tipos de protocolos. Essa biblioteca atualmente suporta os protocolos *https, ftp, gopher, telnet, dict, file* e *ldap* e também suporta certificados HTTPS, HTTP POST, HTTP PUT, upload via FTP, upload HTTP por formulário, proxies, cookies, e autenticação com usuário e senha.

Para a chamada cUR e a API *Tone Analyzer*, primeiramente a frase escutada pelo *Web Speech API* (2017) é transformada em JSON e é informado a URL da API *Tone Analyzer*. Após isso, o cURL é aberto e enviado como parâmetro POST à frase, assim como usuário e senha fornecidos pelo site da IBM Watson. No *VictroRobot* a análise de tom é muito importante, pois ela ajudará a controlar o humor do *chatbot*, tornando-o mais “humano” quando se trata de conversação, ou seja, ao falar algo para o *chatbot* ele enviará para o tone analyzer e com base no retorno fará a alteração de humor para a próxima resposta.

3.8 DialogFlow

Acessando o *dialogflow.com* é necessário acessar o console no menu superior, ao lado da caixa de busca. Será aberta uma nova página para que o usuário faça o *login*, para isso é necessário que o usuário do *dialogflow.com* tenha uma conta *Google*.

O referido console é dividido em 3 áreas: menu lateral esquerdo onde é possível navegar pelas páginas do console, área central, onde o conteúdo é carregado; e o menu lateral direito, onde é possível realizar testes no *chatbot* criado.

4 CONCLUSÕES

Com a visão de compreender o motivo do uso da tecnologia que seria empregada no presente trabalho e também algo que traria vantagem para o uso do *chatbot* desenvolvido, foi percebido que os grandes nomes do mercado como Google, Apple, Amazon entre outros, estão em constante evolução e aprimoramentos. As referidas empresas monitoram os desejos dos usuários para esses aprimoramentos e tornam o sistema ainda mais interativo, dificultando assim o desenvolvimento de algo que ainda não foi desenvolvido por alguma marca atual do mercado.

Visto que a grande maioria dos concorrentes ainda não implantaram um reconhecimento de tom para categorizar humores em seus assistentes, essa característica foi tomada como desafio de implementação do *chatbot VictroRobot*, objeto deste estudo. Houve uma grande redução do tempo de desenvolvimento do software por conta do framework *VictroBrain*, que em sua estrutura conta com funções básicas prontas para os desenvolvedores, como o tratamento dos dados para consultas ao banco de dados e toda a sua padronização MVC, que facilita a programação e a padroniza.

Durante o desenvolvimento houve a necessidade de troca de tecnologia de reconhecimento de voz afim de deixar o *chatbot* o mais rápido possível nas suas respostas, para que a interação com o usuário seja ainda mais imersa e a percepção de estar conversando com um robô seja menor. Também houve a necessidade de atualizações no framework, já que o mesmo foi criado para dar base a este projeto.

Deste modo, pôde-se concluir que todo o processo de desenvolvimento atingiu o objetivo esperado, que era desenvolver um *chatbot* programável com reconhecimento de voz e de tons (humores) utilizando um framework web.

5 REFERÊNCIAS BIBLIOGRÁFICAS

CHROMIUM. Site Oficial. Disponível em: <<https://www.chromium.org/home>>. Acesso em 12 Out. 2017.

FERREIRA, L. P.; UCHÔA, Joaquim Quinteiro. **Desenvolvimento de um *chatbot* para auxiliar o ensino de Espanhol como Língua Estrangeira.** 2008. Disponível em: <http://repositorio.ufba.br/bitstream/1/9629/1/ARTIGO_Desenvolvimento%20de%20um%20chatbot%20para%20auxiliar%20o%20ensino%20de%20espanhol%20como%20l%C3%ADngua%20estrangeira.pdf>. Acesso em: 26 Ago. 2017.

GUPTA, S. et al. An E-Commerce Website based Chatbot. **International Journal of Computer Science and Information Technologies**, v. 6, n. 2, p. 1483-1485, 2015. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.734.8303&rep=rep1&type=pdf>>. Acesso em: 28 Ago 2017.

MINETTO, E. L. **Frameworks para Desenvolvimento em PHP**. São Paulo: Novatec, 2007. Disponível em: <<http://www.martinsfontespaulista.com.br/anexos/produtos/capitulos/243418.pdf>>. Acesso em: 30 Ago 2017.

RATO, J. P. C. **Conversação homem-máquina. Caracterização e avaliação do estado actual das soluções de speech recognition, speech synthesis e sistemas de conversação homem-máquina**. 2016. Tese de Doutorado.). Disponível em: <<https://iconline.ipleiria.pt/bitstream/10400.8/2375/1/jo%C3%A3o%20Rato-Mestrado%20em%20Eng.Inform%C3%A1tica-Computa%C3%A7%C3%A3o%20M%C3%B3vel.pdf>>. Acesso em: 29 Ago 2017.

SELLITTO, M. A. **Inteligência artificial: uma aplicação em uma indústria de processo contínuo**. *Gestão & Produção*, v. 9, n. 3, p. 363-376, 2002. Disponível em: <http://www.scielo.br/pdf/%0D/gp/v9n3/14574.pdf> >. Acesso em: 24 Nov. 2017.

SIMIONI, M. C.; BETINI, R. C. **Monitoramento Da Frequência Cardíaca Via Método De Magnificação De Vídeo Euleriana**, 2014. Disponível em: <http://www.canal6.com.br/cbeb/2014/artigos/cbeb2014_submission_703.pdf >. Acesso em: 24 Nov. 2016.

WEB SPEECH API – **Documentation**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API >. Acesso em: 10 Out. 2017.