

# CONSTRUÇÃO DE UM *CHATBOT* DE APOIO À CONSTRUÇÃO DE LISTA DE TAREFAS INSERIDO EM UM APLICATIVO DE MENSAGENS

## DESIGNING A CHATBOT FOR SUPPORT THE TO-DO LIST INSERTED IN A MESSAGE APPLICATION

Mateus Canavez Testoni\*

Adinovam Henriques de Macedo Pimenta\*\*

### RESUMO

Este trabalho apresenta o processo de desenvolvimento de um *chatbot* inserido em um aplicativo de mensagens com o objetivo de dar apoio ao usuário na construção de uma lista de tarefas. Nele está descrito um breve histórico dos *chatbots* pioneiros e mais relevantes no Brasil e no mundo, além de abordar as vantagens e uma visão mercadológica do desenvolvimento de *chatbots*.

**Palavras-chave:** *Chatbot*. Aplicativos de mensagens. Lista de tarefas.

### ABSTRACT

This paper presents the process of developing a chatbot inserted in a messaging application designed to support the user on building a to do list. A brief history about the pionerr and most relevant chatbots in Brazil and the world is described. This paper also addresses the advantages and a market view of the chatbot development.

**Keywords:** Chatbot. Messaging apps. To-do list.

### Introdução

Uma linguagem natural é aquela usada como meio de comunicação entre os seres humanos como, por exemplo, o inglês, o francês, o alemão, entre outras. Todas essas linguagens possuem gramática e sintaxe que, embora possam possuir ambiguidades, seguem princípios de otimização da comunicação do seu meio. As línguas naturais evoluíram em conjunto atendendo as novas necessidades da humanidade (FINATTO; LOPES; CIULLA, 2015).

---

\* Graduação em Ciência da Computação. Faculdade de Tecnologia, Ciências e Educação (FATECE). [mctestoni@gmail.com](mailto:mctestoni@gmail.com)

\*\* Docente da Faculdade de Tecnologia, Ciências e Educação (FATECE) e Orientador desta pesquisa. [adinovam@icmc.usp.br](mailto:adinovam@icmc.usp.br)

A linguagem formal, em contrapartida, tem sua utilidade na transferência precisa de informação, não tendo espaço para ambivalências ou equívocos. As notações matemáticas e as linguagens de programação são exemplos de linguagens formais (LOPES; VIEIRA, 2010; DIAS-DA-SILVA, 1996).

*Chatbots* são programas de computador que se comunicam com pessoas por meio da linguagem natural utilizando mensagens de texto ou até mesmo mensagens de voz inseridos em aplicativos projetados com este propósito. Um *chatbot* pode ser desde um receptor de simples comandos via *chat*, até um robô virtual inteligente que interage com o ser humano, sendo capaz de compreender e responder o que o usuário digitar ou falar.

Por muitos anos o principal objetivo dos *chatbots* foi produzir uma conversa interessante, confundindo o ser humano a ponto de fazê-lo acreditar que está conversando com outro ser humano. Desde que Turing propôs o jogo da imitação (TURING, 1950), foram construídos vários programas com este propósito, tais como o ELIZA (WEIZENBAUM, 1966), o PARRY (COLBY, 1973), o RACTER (CHAMBERLAIN, 1984) e o ALICE (WALLACE, 2009). Apesar das tentativas de tornar a conversa desenvolvida com estes *chatbots* a mais próxima possível das conversas desenvolvidas por humanos, os resultados não conseguiram satisfazer por completo os usuários. Isso deve-se ao fato de que as regras de conversação eram insuficientes para tornar as conversas mais naturais (HILL; FORD; FARRERAS, 2015).

Os aplicativos (*apps*) de mensagens são ambientes propícios para a interação de *chatbots* com o público e cresceram muito nos últimos anos, passando as redes sociais em número de usuários ativos por mês, um número crescente de jovens adultos está migrando para os *apps* de mensagens para sua comunicação do dia a dia (JAIN; JAIN, 2017).

Aplicativos móveis (*apps*), *sites* e outros *softwares* comuns são produtos consolidados no mercado atual, mas todos eles têm um custo. Tomando como exemplo os *apps*, para cada loja existe um aplicativo, para cada aplicativo existe uma plataforma diferente, para cada plataforma temos um ambiente de desenvolvimento, para cada ambiente de desenvolvimento temos um *design* e para cada *design* um comportamento diferente. O *chatbot* é um produto que traz a oportunidade de se desvencilhar de grande parte dessas complicações, pois ele segue a universalidade das aplicações de mensagem que funcionam em várias plataformas e também em vários dispositivos.

O objetivo deste trabalho é criar um *chatbot* de apoio à construção de lista de tarefas inserido em um aplicativo de mensagens. O programa vai identificar, a partir das

mensagens do usuário, as tarefas que deve adicionar ou apagar de sua lista. Para atingir este objetivo, pretende-se utilizar técnicas de análise léxica para identificar por meio das mensagens, o que o usuário deseja que seja executado em sua lista de tarefas.

Com a ajuda do *chatbot* a ser desenvolvido neste trabalho, será possível ter acesso a tarefas salvas junto a um assistente virtual presente em qualquer aplicativo de mensagens.

## **1 Histórico**

Um dos primeiros *chatbots* mais famosos é o ELIZA (WEIZENBAUM, 1966), que tornava possível alguns tipos de conversas entre homem e máquina, simulando um psicoterapeuta. Seu funcionamento era basicamente decomposição de frases e análise de palavras-chaves que eram gatilhos para respostas de ELIZA de sua base de dados.

Entre os *chatbots* mais importantes da história estão o PARRY (COLBY, 1973) que simulava um paciente com esquizofrenia; o RACTER, desenvolvido inicialmente por Chamberlain (1984) que, segundo descreve o autor, o programa escreveu sozinho o livro “*The Policeman’s Beard is Half Constructed*”; e o ALICE (WALLACE, 2009), baseado na linguagem de marcação para IA, o AIML, também criado por Wallace. O ALICE foi um dos primeiros *chatbots* desenvolvidos para interação na internet e seu nome é uma sigla que significa "Linguística artificial em entidade computacional na Internet".

No Brasil, os pioneiros mais famosos da área foram o *Se7e Zoom*<sup>1</sup> em 2001 e o Robô Ed<sup>2</sup> em 2003. O primeiro foi desenvolvido para a empresa Unilever com o intuito de ser representante da marca de pastas de dente *Close-Up*, interagindo com os usuários acerca de qualquer assunto e tirando dúvidas específicas da empresa (MOURA, 2003). Este foi um dos primeiros *chatbots* a conversar através do programa de mensagens instantâneas ICQ<sup>3</sup>; o Robô Ed foi desenvolvido pelo Programa Nacional de Racionalização do Uso dos Derivados de Petróleo e do Gás Natural, interagindo sobre assuntos relacionados ao meio ambiente e tem sua ligação com a Petrobras e o Ministério de Minas e Energia.

---

<sup>1</sup> <http://www.inbot.com.br/novo/cases.php#sete>

<sup>2</sup> <http://www.ed.conpet.gov.br/br/converse.php>

<sup>3</sup> <https://icq.com/windows/pt>

## 2 Cenário atual

Grandes empresas seguiram desenvolvendo soluções em software baseadas na ideia de assistentes virtuais e outros elementos de inteligência artificial com o passar dos anos, como por exemplo os assistentes virtuais *Siri*<sup>4</sup> da *Apple* em 2011, *Google Now*<sup>5</sup> da *Google* em 2012, *Cortana*<sup>6</sup> da *Microsoft* em 2014 e *Alexa*<sup>7</sup> da *Amazon* em 2014. Todos esses têm em comum o objetivo de facilitar a vida do usuário dando a ele a possibilidade de realizar tarefas de maneira mais intuitiva, como telefonar para alguém, fazer uma busca na internet, abrir documentos virtuais ou até ligar e desligar aparelhos domésticos com um simples comando de voz ou uma mensagem de texto.

O *Facebook* lançou a sua integração com *chatbots* em seu aplicativo de mensagens, o *Facebook Messenger*<sup>8</sup>, em abril de 2016. No final do mesmo ano contavam com 30.000 *chatbots* na plataforma (JAIN; JAIN, 2017). Vários outros aplicativos de mensagens também fornecem integração com *chatbots*, permitindo que o usuário interaja de diversas maneiras, seja para conseguir alguma informação específica da empresa, sanar dúvidas de produtos ou até mesmo para fazer compras e adquirir serviços de forma automatizada.

## 3 Mercado

Os aplicativos de mensagens cresceram muito nos últimos anos, passando as redes sociais em número de usuários ativos por mês conforme pode-se observar na Figura 1. Esses dados mostram que um número crescente de jovens adultos está migrando das grandes redes sociais, como o *Facebook*, para micro-redes (*micro-networks*) privadas como o *Facebook Messenger*, *Slack*, *WeChat* e *Telegram* para sua comunicação do dia a dia. Uma mudança massiva de comportamento, pois até então as redes sociais eram consolidadas como meio principal de interação entre os usuários e, como consequência disso, era o ambiente determinante para o marketing digital de empresas (JAIN; JAIN, 2017).

---

<sup>4</sup> <https://www.apple.com/br/ios/siri>

<sup>5</sup> <https://www.google.com/intl/pt-BR/landing/now>

<sup>6</sup> <https://support.microsoft.com/pt-br/help/17214/windows-10-what-is>

<sup>7</sup> <https://www.amazon.com/meet-alexa/b?ie=UTF8&node=16067214011>

<sup>8</sup> <https://developers.facebook.com/docs/messenger-platform>

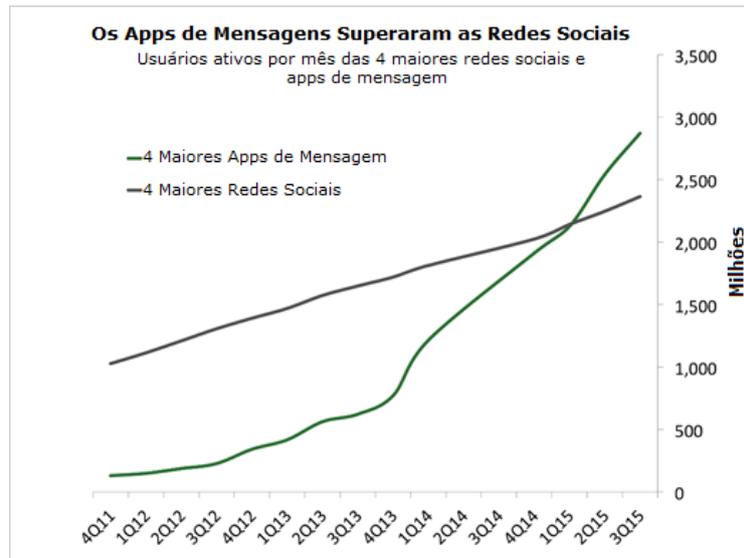


Figura 1: Número de usuário de Apps de Mensagem Superaram e Redes Sociais  
**Fonte:** <http://www.businessinsider.com/the-messaging-app-report-2015-11>

As empresas demonstram que estão observando essa migração para essas *micro-networks*, pois já estão investindo em plataformas que facilitam a construção e publicação de um *chatbot*, como por exemplo o *Microsoft Bot Platform*<sup>9</sup> que permite a criação de *chatbots* com interpretação e sentidos cognitivos mais sofisticados; o *Facebook Messenger Platform* que permite a criação e publicação de um *chatbot* em um dos aplicativos de mensagens mais utilizados do mundo; o *Telegram Bot API*<sup>10</sup> que permite a construção de *chatbots* que atuem em chats em grupo, participando da interação coletiva dos usuários; entre outros com funções parecidas.

#### 4 Vantagens

Além do mercado estar demonstrando um grande desenvolvimento nessa área, existem vantagens na utilização de *chatbots* que reforçam a importância que vem ganhando no cenário computacional, mas que também coloca os *bots* em uma disputa contra outras linhas de investimento em tecnologia. Essas linhas são formadas principalmente por aplicações *WEB (sites)* e aplicações *mobile (apps)*.

Os *sites* foram por um bom período o único item necessário para a presença de uma empresa no meio digital. Com o advento dos *apps* em conjunto com a popularidade

<sup>9</sup> <https://dev.botframework.com/>

<sup>10</sup> <https://core.telegram.org/bots/api>

gigantesca dos *smartphones* no mundo, os investimentos em aplicações *WEB* caíram drasticamente devido aos benefícios que a tecnologia *mobile* trazia para suas aplicações. O *chatbot*, por sua vez, possui vantagens claras quando comparado diretamente com *sites* e *apps*:

- **Interface familiar e intuitiva:** Como o usuário interage com o *bot* em um aplicativo de mensagem que ele já conhece, a utilização torna-se muito menos complicada, ao passo que os *apps* para celular possuem a sua própria interface;
- **Menor consumo de memória:** Não é necessário baixar um *chatbot*. Isso evita todo o carregamento de imagens de um *site* e também evita o download de um aplicativo;
- **Multiplataforma:** Seja qual for o dispositivo, sistema operacional ou navegador, o *chatbot* vai funcionar, contanto que tenha suporte ao *messenger* no qual ele foi construído. Não é o mesmo com os *apps* que tem uma especificação diferente para sistemas operacionais diferentes.

## **5 Inserção do Chatbot em um Aplicativo de Mensagens**

Nesta seção serão abordadas as etapas de desenvolvimento do *chatbot* inserido no aplicativo de mensagens *Facebook Messenger* que funcionará como apoio à construção de lista de tarefas. A partir deste ponto, vamos nos referir ao *chatbot* aqui desenvolvido como *Taskbot*.

As etapas de desenvolvimento do *Taskbot* vão desde a preparação do ambiente do servidor e a conexão com o *Facebook Messenger*, até a implementação de suas funções, bem como do banco de dados necessário ao seu funcionamento.

### **5.1. Preparação do Ambiente**

Os *chatbots* inseridos em aplicativos de mensagens utilizam um servidor *web* para processar mensagens recebidas e enviar as respostas adequadas. A plataforma escolhida para hospedar o projeto foi o *Heroku*, por oferecer um serviço gratuito com suporte para a plataforma de desenvolvimento *Node.js*, em que o *Taskbot* foi codificado.

Um arquivo deve ser adicionado à pasta do projeto para comunicar ao servidor criado no *Heroku* o que ele deve executar. O nome desse arquivo é *Procfile* e não possui

nenhuma extensão. Para o projeto foi necessário especificar no *Procfile* que se tratava de um servidor web, rodando em *Node.js* e o arquivo índice *index.js*, como pode ser verificado na Figura 2.



```

Procfile
1 web: node index.js
    
```

Figura 2: Conteúdo do arquivo *Procfile*.

## 5.2. Conexão com o Aplicativo de Mensagens

O aplicativo de mensagens escolhido para executar as funções do *Taskbot* foi o *Facebook Messenger*, devido à grande popularidade do *app* e ao suporte completo que é dado para a criação de *chatbots*. Para que se tenha um *chatbot* funcional através dessa plataforma é necessário criar um *Facebook App*. A criação desse *app* foi realizada na seção de desenvolvedores do site do *Facebook*<sup>11</sup>.

Para a conexão entre o servidor criado e o *Facebook App*, foi necessário fornecer ao *app* um *webhook*, que é o endereço do servidor *Heroku*, o *token* descrito no código do arquivo *index.js* e as permissões do *chatbot*, como pode ser observado na Figura 3.



Figura 3: Configuração de *webhook* dentro de um *Facebook App*.

Foi criada anteriormente uma página no *Facebook* chamada *Taskbot*, que vai funcionar como um perfil do *chatbot* no aplicativo. É necessário atrelar a página ao *app*

<sup>11</sup> <https://developers.facebook.com>

para que seja gerado outro *token* colocado no código do arquivo *index.js* do projeto, finalizando a conexão do servidor com o aplicativo de mensagens.

### **5.3. Implementação**

O projeto contém uma biblioteca *Redis*<sup>12</sup> que permite manipular um banco de dados *Redis*<sup>13</sup> em um código na plataforma *Node.js*. *Redis* significa *Remote Dictionary Server*, que é um banco de dados não-relacional e os dados são armazenados na forma chave-valor, isto é, os dados são separados por chaves que podem ser simples *strings* e são usadas nas funções como parâmetro para as funções do banco. No projeto é utilizado o *ID* do usuário fornecido pelo *Facebook Messenger* como chave para separar os dados de cada usuário. Esses dados englobam tarefas adicionadas e comandos que ativam as principais funções do *Taskbot*.

O *Redis* foi escolhido por ser um banco de acesso rápido e ter uma boa afinidade com aplicações em *Node.js*. O que se mostrou necessário devido ao grande número de acessos que o *chatbot* precisa fazer para realizar suas funções corretamente e em tempo hábil para a comunicação com o usuário.

Foram implementadas funções de adicionar, deletar e listar tarefas, sendo que todas estas tarefas acessam o banco de dados e recuperam a lista completa de tarefas para executar a ação do comando. Além destas tarefas, também foi implementada a função de mudar um dos comandos de texto entendidos pelo *chatbot*, tornando o programa customizável pelo usuário. Nesse caso apenas a seção de comandos do usuário é recuperado do banco.

As funções se baseiam em técnicas de análise léxica para identificar o comando digitado pelo usuário, além de garantir respostas corretivas caso o usuário informe o comando de maneira incorreta.

## **6 Resultados**

Quando um usuário do *Facebook Messenger* fala pela primeira vez com o *Taskbot*, uma mensagem de boas-vindas com comandos pré-programados é exibida ao usuário, conforme pode ser observado na Figura 4.

---

<sup>12</sup> <https://www.npmjs.com/package/redis>

<sup>13</sup> <https://redis.io/>

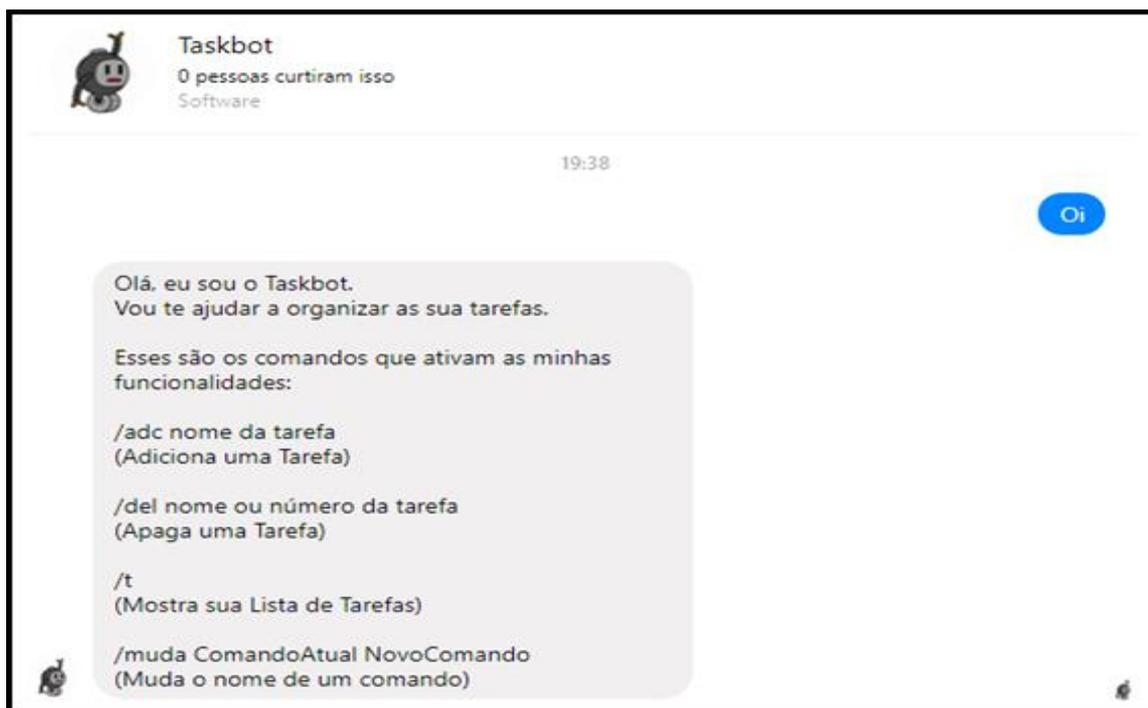


Figura 4: Primeira interação com o usuário

O comando de adicionar uma tarefa nova retorna uma mensagem que mostra se a adição teve êxito. Logo em seguida o número de tarefas armazenadas na lista também é exibido conforme pode-se ver na Figura 5. O *chatbot* também pode retornar uma mensagem com uma correção, caso o usuário tenha usado o comando de maneira errada ou faltando parâmetros.

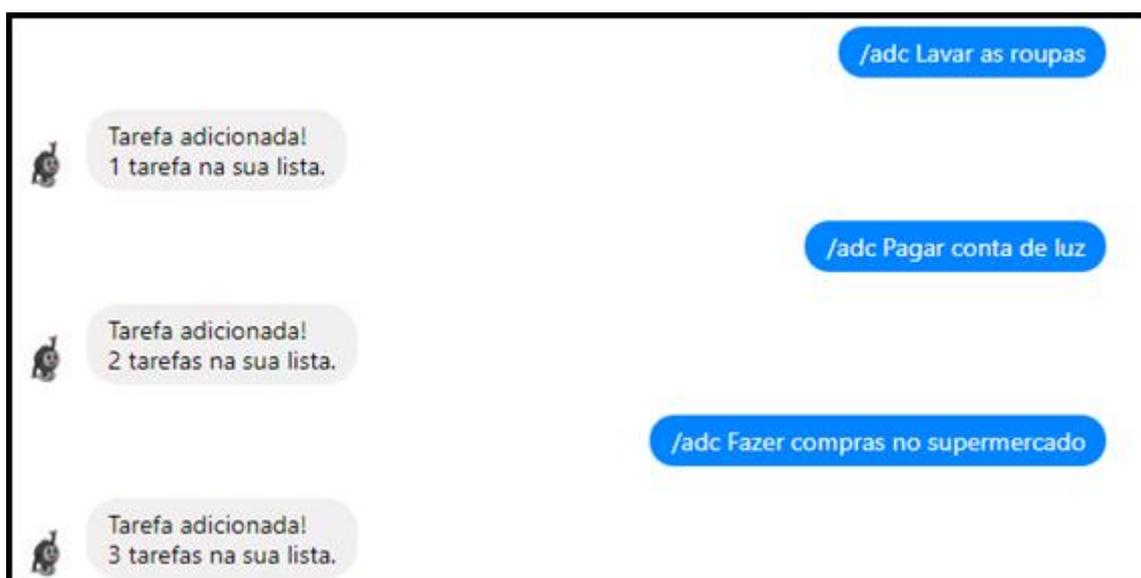


Figura 5: Adicionando tarefas

A forma de visualizar as tarefas adicionadas é a utilização de um comando responsável por listar tarefas. Como pode ser visto na Figura 6, essa função não precisa de parâmetros e retorna para o usuário o número de tarefas e uma lista numerada com as tarefas adicionadas na ordem em que foram incluídas no banco.

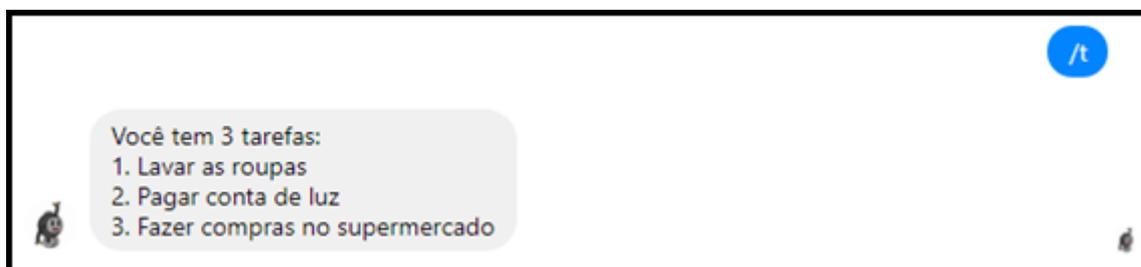


Figura 6: Listando tarefas

A função de deletar as tarefas funciona ou passando o nome da tarefa como parâmetro ou, como pode ser visto na Figura 7, passando apenas a posição que ela se encontra na lista. Em seguida, é retornado para o usuário uma mensagem mostrando o nome da tarefa removida, mas também pode exibir uma mensagem de correção se o usuário usar o comando de forma errada.

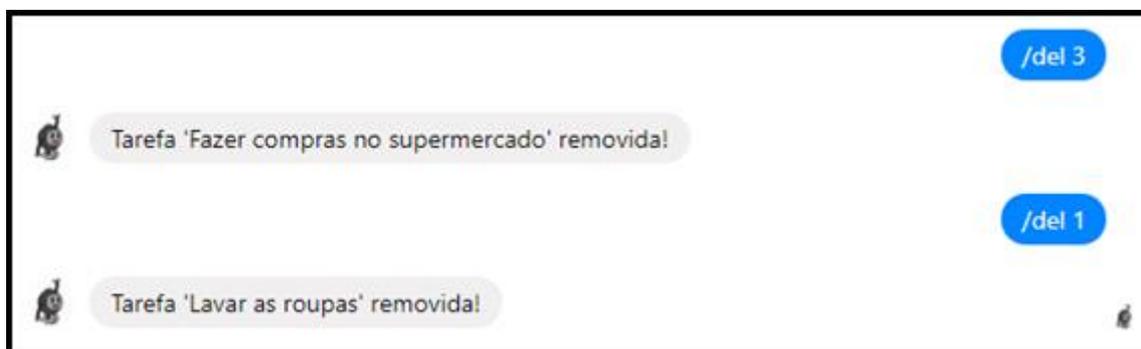


Figura 7: Removendo tarefas

A função de mudar comando pode mudar qualquer comando citado anteriormente, além de poder mudar a si mesmo. Como pode ser visto na Figura 8, com esse comando o *chatbot* retorna uma mensagem mostrando qual comando foi alterado e qual é o nome do novo comando. Além disso, o *chatbot* não permite que este novo comando tenha o mesmo nome de outro comando já existente. Caso o usuário tente fazer isso, uma mensagem será retornada avisando que a operação é inválida.

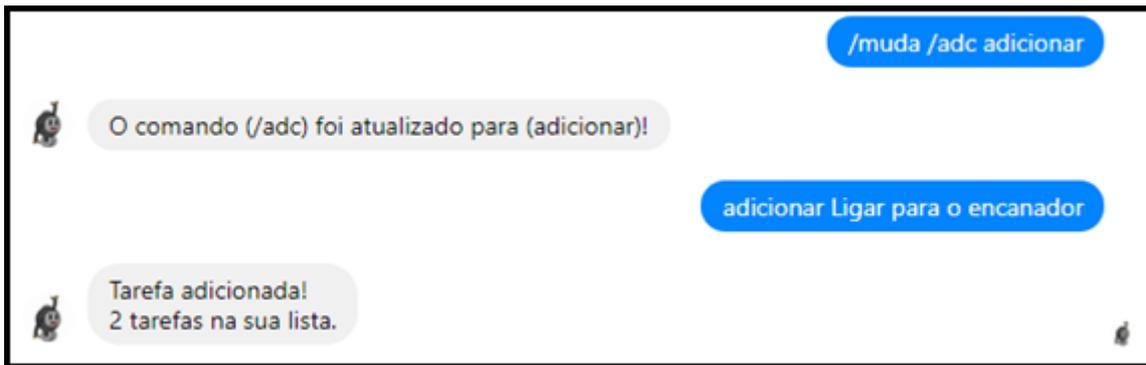


Figura 8: Customizando comandos

### Considerações Finais

Com o *chatbot* desenvolvido atingiu-se o objetivo proposto de conectar um *chatbot* a um aplicativo de mensagens com uma grande base de usuários ativos; dar suporte ao usuário para a construção e manutenção de uma lista de tarefas; mostrar que seu uso é vantajoso perante aplicativos com a mesma finalidade, visto que o usuário que já possui o aplicativo não precisa baixar nenhum conteúdo a mais; e dar destaque ao comportamento multiplataforma que acompanha o próprio aplicativo de mensagens que já é desenvolvido para diferentes tipos de sistemas operacionais.

Como trabalhos futuros, pretende-se: acrescentar um sistema de alarme que permita ao usuário agendar e ser avisado de suas tarefas agendadas; implementar técnicas mais avançadas de processamento de linguagem natural para tornar a experiência do usuário melhor, evitando desentendimentos e mau uso do *chatbot*; e adaptar o código para as APIs de outros aplicativos de mensagens, tornando o *chatbot* acessível a mais perfis de público; por fim serão realizados testes com usuário para verificar a acurácia e a usabilidade do *chatbot*.

### Referências

CHAMBERLAIN, W. **The Policeman's Beard is Half Constructed**. New York: Warner Books, 1984.

DIAS-DA-SILVA, B. C. **A fase tecnológica dos estudos da linguagem: o processamento automático das línguas naturais**. 1996. 274 p. Tese (Doutorado em Linguística e Língua Portuguesa) – Faculdade de Ciências e Letras, Universidade Estadual Paulista, Araraquara, 1996.

HILL, J.; FORD, W. R.; FARRERAS, I. G. Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. **Computers in Human Behavior**, v. 49, p. 245-250, 2015.

JAIN, N.; JAIN, A. **A Survey on Popularity of Chat-Bots**. International Conference On Emanations in Modern Technology and Engineering, Rajasthan, v. 5, n. 3, 2017.

LOPES, L.; VIEIRA, R. Processamento de Linguagem Natural e o Tratamento Computacional de Linguagens Científicas. In: PERNA, C. L.; DELGADO, H. K.; FINATTO, M. J. (Org.). **Linguagens Especializadas em Corpora: modos de dizer e interfaces de pesquisa**. Porto Alegre: EdiPucrs, 2010. p. 183-201.

MCKITTERICK, W; BI INTELLIGENCE. **Messaging apps are now bigger than social networks**. Disponível em: <<http://www.businessinsider.com/the-messaging-app-report-2015-11>>. Acesso em: 22 set. 2017.

MOURA, T. **Um Chatterbot para aquisição automática de perfil do usuário**. 2003. 127 p. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2003.

SHEJY, N.; THAKUR, S.; JABSHETTY, M. **Chatbots on Messaging Platform Revolution in app world**. International Conference On Emanations in Modern Technology and Engineering, Rajasthan, v. 4, n. 6, 2017.

TURING, A. M. **Computing machinery and intelligence**. *Mind*, v. 59, n. 236, p. 433-460, 1950.

VIEIRA, T, et al. **Chatterbots como colaboradores nos serviços de referência: uma apresentação das potenciais atividades do robô nos serviços de informação na biblioteca universitária**. 2017. 52 p. Trabalho de Conclusão de Curso (Graduação em Biblioteconomia) – Faculdade de Biblioteconomia, Universidade Federal do Pará, Belém, 2017.

WALLACE, R. **The Anatomy of A.L.I.C.E**. Parsing the Turing test: Philosophical and methodological issues in the quest for the thinking computer. Springer, 2009, p.181-210.

WEIZENBAUM, J. ELIZA—a computer program for the study of natural language communication between man and machine. **Communications of the ACM**, v. 9, n. 1, p. 36-45, 1966.