

## ITGuy: Solução de *Chatbot* para *Service Desk*

Adinel de Deus Florentino dos Santos<sup>1</sup> – Fatec Carapicuíba

Fernando Florencio de Oliveira<sup>2</sup> – Fatec Carapicuíba

Lucas dos Santos Silva<sup>3</sup> – Fatec Carapicuíba

Michel Gonçalves Branco Ferreira<sup>4</sup> – Fatec Carapicuíba

Pedro Sergio Vaz<sup>5</sup> – Fatec Carapicuíba

Profa.Me Magali Amorim Mata<sup>6</sup> – Fatec Carapicuíba

### RESUMO

Algo muito comum na atualidade são os serviços de mensageria: amplamente utilizados os *chats* têm as mais diversas aplicações, seja para uma conversa informal ou para um atendimento a um usuário, onde o usuário insere uma mensagem, geralmente uma dúvida, e o *chat* o responde podendo ter ou não um ser humano por trás da resposta. Este artigo tem como objetivo geral relatar o desenvolvimento de uma aplicação de *Chatbot* para auxiliar na resolução de requisições de serviço de primeiro nível em um *Service Desk*. Trata-se de uma pesquisa de caráter exploratório e experimental e de uma pesquisa aplicada com abordagem descritiva e qualitativa. O desenvolvimento do *Chatbot* baseia-se na linguagem de programação C#. Para a construção do banco de dados utiliza-se o MySQL e para a interpretação de mensagens dos usuários, processamento de linguagem natural, utilizada a *Application Programming Interface* (API) e a *Language Understand Intelligence Service* (LUIS) da Microsoft.

**Palavras-chave:** *Chatbot*, Processamento de Linguagem Natural, *Service Desk*.

### ABSTRACT

*Something very common nowadays are the messaging services: widely used chat rooms have the most diverse applications, whether for an informal conversation or for a service to a user, where the user inserts a message, usually a doubt, and the chat answers him/her whether there is or not a human being behind the answer. This article aims to report the development of a Chatbot application to assist in the resolution of first level service requests in a Service Desk. It is an exploratory and experimental research and an applied research with a descriptive and qualitative approach. The development of Chatbot is based on the C # programming language. For the construction of the database it was used MySQL and for the interpretation of messages of the users, natural language processing, used the Application Programming Interface (API) and the Language Understand Intelligence Service (LUIS) of Microsoft.*

**Keywords:** *Chatbot*, Natural Language Processing, *Service Desk*.

---

<sup>1</sup> - Discente do CST em Análise e Desenvolvimento de Sistemas – *e-mail*: adinel.santos@fatec.sp.gov.br

<sup>2</sup> - Discente do CST em Análise e Desenvolvimento de Sistemas – *e-mail*: fernando.oliveira57@fatec.sp.gov.br

<sup>3</sup> - Discente do CST em Análise e Desenvolvimento de Sistemas – *e-mail*: lucas\_san20@hotmail.com

<sup>4</sup> - Discente do CST em Análise e Desenvolvimento de Sistemas – *e-mail*: michel.ferreira2@fatec.sp.gov.br

<sup>5</sup> - Discente do CST em Análise e Desenvolvimento de Sistemas – *e-mail*: pedro\_vazz@hotmail.com

<sup>6</sup> - Mestre em Gestão e Desenvolvimento da Educação Profissional pelo CEETEPS – *e-mail*: magali.mata@fatec.sp.gov.br

## 1 INTRODUÇÃO

O ser humano sempre buscou substituir a força de trabalho braçal por algo mais automatizado, tornando os processos muito mais rápidos e com isso várias tecnologias foram surgindo, sejam máquinas robustas ou *softwares* complexos. Algo muito comum atualmente são os serviços de mensageria: amplamente utilizados os *chats* têm as mais diversas aplicações, seja para uma conversa informal ou para um atendimento a um usuário, onde o usuário insere uma mensagem, geralmente uma dúvida e o *chat* o responde podendo ter ou não um ser humano por trás da resposta. Um exemplo desse tipo pode ser visto em *sites* como o *Facebook*, onde empresas que utilizam a rede colocam serviços de *chatbot* para responder seus clientes nos momentos em que esses possuem alguma dúvida, e quando a questão não é solucionada o usuário geralmente é direcionado para um outro local onde a dúvida possa ser sanada.

Atualmente os *chatbots* estão cada vez mais presentes na vida dos seres humanos. Tratam-se de assistentes pessoais, como a *Siri* ou a *Cortana*, que estão se popularizando entre os usuários de *smartphones* e computadores pessoais, tendo em vista que essas tecnologias vêm se desenvolvendo de uma forma muito acelerada trazendo inúmeros benefícios e inovações na vida dos indivíduos. Além de auxiliar no agendamento de tarefas, realização de pesquisas e até mesmo para entretenimento através de diálogos descontraídos, essas tecnologias estão sendo usadas no mundo corporativo para ajudar os usuários a obter as informações desejadas de maneira mais rápida e interativa, como exemplo temos a aplicação *Eva da Ticket* que conversa com os usuários para apoiar na navegação dentro do *site*.

Esses são exemplos de *chatbots* destinados aos usuários finais de diversas tecnologias ou para clientes externos de determinadas empresas. Entretanto, é possível utilizar essa tecnologia para apoiar os processos internos de uma Organização? Para isso é necessário entender-se o funcionamento dos *bots*, em quais processos seria possível utilizá-los e como essa implementação se encaixaria em determinado contexto.

Desta forma, a partir desse questionamento, a pesquisa tem como objeto de estudo a área de *Service Desk*, mais especificamente de primeiro nível, como processo a ser automatizado pelo sistema já que essa área atua com um modelo similar ao da aplicação: os usuários fazem perguntas ou solicitações que são atendidas pontualmente pelos funcionários da área de suporte técnico. Desta forma, o artigo tem como objetivo geral relatar o

desenvolvimento de uma aplicação de *Chatbot* para auxiliar na resolução de requisições de serviço de primeiro nível em um *Service Desk*. Os objetivos específicos são: conceituar e entender o funcionamento de um *Service Deks*, *Chatbots* e processamento de linguagem natural.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 *Service Desk*

Segundo Cavalari & Costa (2005), *Service Desk* é um sistema que pertence à estrutura organizacional, constituindo-se em um departamento específico para onde são direcionadas perguntas e aonde espera-se que problemas sejam solucionados. Atualmente, segundo os autores, o objetivo dos sistemas de *Service Desk* é assumir o papel de um agente facilitador de informações para o usuário de uma dada instituição, sejam essas informações de natureza técnica computacional ou não. Assim, o conceito de *Service Desk* ultrapassa a conceituação de suporte técnico.

Segundo Medeiros e Soares (2010), em qualquer empresa, um usuário de tecnologia da informação pode passar por diversas dificuldades técnicas ou processuais no decorrer do desenvolvimento das suas atividades e, quando não solucionadas, essas dificuldades podem impactar em sua produtividade. Para evitar ou mitigar esse impacto negativo existe nas corporações um setor que é responsável por receber todas as solicitações de resolução para esses problemas para resolve-los ou, se for o caso, direciona-los ao solucionador. Esse setor é chamado em algumas organizações de *help desk*.

Ainda de acordo com os autores, o *Service Desk* é o centralizador de todos os incidentes gerados pelos usuários, dessa forma é de responsabilidade do setor manter os solicitantes informados do andamento de seus chamados e controlar a resolução dos mesmos. Sendo assim o *Service Desk* é a *interface* entre os clientes internos e os serviços de TI, sejam eles para resolução de problemas técnicos ou de execução de processos internos.

O atendimento prestado pelo *Service Desk* se baseia na estrutura de perguntas e respostas, em que as perguntas são realizadas pelos usuários e as respostas são as resoluções ou informações providas pelos analistas do *Service Desk* (CAVALARI e COSTA, 2005).

De acordo com Cohen (2008) o *Service Desk* pode ser dividido em três níveis de atendimento. O primeiro nível é onde se concentram a resolução de problemas mais simples,

requerem um menor nível de *expertise*, são aqueles que já são conhecidos e recorrentes, isto é, relativos a problemas que podem ser resolvidos no momento do atendimento e, quando não é possível, faz-se o direcionamento ao segundo e terceiro nível de acordo com a complexidade da requisição. Se uma requisição não pode ser atendida no primeiro nível, ela seguirá para o segundo nível e posteriormente ao terceiro nível, onde a *expertise* necessária para resolução é mais alta.

Dentro do primeiro nível de suporte, existem duas formas principais de atendimento: o atendimento solucionador e o direcionador. O solucionador resolve problemas à medida que eles surgem, geralmente problemas simples que podem ser resolvidos no momento do atendimento que na maioria das vezes é via telefone. Já o direcionador encaminha o problema a um grupo solucionador específico de maior nível de conhecimento quando não é possível a resolução desse problema no primeiro nível.

A maioria das requisições feitas a um *Service Desk* são solucionadas no primeiro nível. Assim, é conveniente ter-se uma solução de *chatbot* que seja capaz de interagir com os usuários através de mensagens de texto para sugerir soluções para os problemas conhecidos e mapeados dentro do suporte desse primeiro nível, exercendo o papel de solucionador ou para direcionar aos níveis superiores quando não encontrar uma solução na base de conhecimento, fazendo o papel de direcionador.

## **2.2 Chatbot**

Para Chantarotwong (2005) *chatbot* é um programa de computador capaz de simular um diálogo com um ser humano de forma que aparentemente simule ser uma pessoa do outro lado da conversa. Se esse objetivo for atingido pode-se considerar que o *bot* passou no teste de Turing. Este teste classifica se uma máquina pode ser considerada inteligente se for capaz de manter um diálogo com um ser humano e não ser identificada como máquina. O teste será abordado mais adiante.

Ainda de acordo com Sotto & Costa (2015) os *chatbots* são capazes de responder perguntas ou estabelecer uma conversa utilizando uma tecnologia chamada de Processamento de Linguagem Natural (PLN) e uma base de dados previamente estabelecida pelos seus desenvolvedores. Não se desenvolve, necessariamente, um novo conhecimento a partir dos diálogos, apesar de ser possível utilizando outras metodologias de desenvolvimento para obtenção desse resultado, tal como *machine learning* (aprendizado de máquina).

O PLN torna uma máquina capaz de compreender a fala e/ou escrita e converte-la a uma linguagem inteligível ao computador, quando aplicado a um sistema computacional pode capacitar a máquina a compreender a fala e/ou escrita e então converter em uma linguagem que o computador possa compreender. Para isso, os sistemas de PLN identificam padrões estruturais nas frases a partir de três pilares da comunicação: som (fonologia), estrutura (morfologia e sintaxe) e significado (semântica e pragmática) (PEREIRA, 2012).

Para Ferreira e Uchôa (2005) o *Chatbot* é um programa de computador criado para manter uma conversa com os seres humanos, apesar da definição simples a implementação possui grande nível de complexidade. Para melhor entender-se o funcionamento de um *software* desse tipo, é necessário compreender-se um ramo do conhecimento dentro da inteligência artificial. Inteligência artificial (IA) é uma linha do conhecimento que almeja simular com excelência os aspectos de um ser humano e para isso se utiliza de elementos computacionais. Dentro da IA há diversas ramificações. Para a existência de um *chatbot*, uma ramificação essencial é o PLN (RUSSELL e NORVIG, 2002).

Atrelado a ideia da inteligência artificial surgiu um teste, utilizado como parâmetro para se saber quão perfeita está a simulação que a máquina fez do ser humano. A ideia do teste foi idealizada em 1950 com Alan Turing, como mencionado acima, e ficou conhecido como teste de Turing. O teste consiste em dois ou mais agentes em uma troca de mensagens, onde um desses agentes é um sistema computacional. Quando todos os membros conversam e não conseguem identificar um computador em meio a eles pode-se dizer que a máquina em questão é inteligente e passou no teste de Turing. Todavia, até os dias atuais nenhum computador foi capaz de passar no teste.

### **3 PROCEDIMENTOS METODOLÓGICOS**

Trata-se de uma pesquisa de caráter exploratório e experimental, que busca entender o funcionamento de um *Service Desk* de primeiro nível, como condição ao desenvolvimento e à estruturação de como deverá ocorrer o funcionamento do *software*. São investigados os importantes conceitos e tecnologias que estão diretamente ligados ao *chatbot*, para a construção da aplicação utilizando-se todos os conhecimentos adquiridos.

O desenvolvimento do *Chatbot* baseia-se na linguagem de programação C#. Para a construção do banco de dados utiliza-se o MySQL e para a interpretação de mensagens dos usuários, processamento de linguagem natural, utilizada a *Application Programming Interface* (API) e a *Language Understand Intelligence Service* (LUIS) da Microsoft.

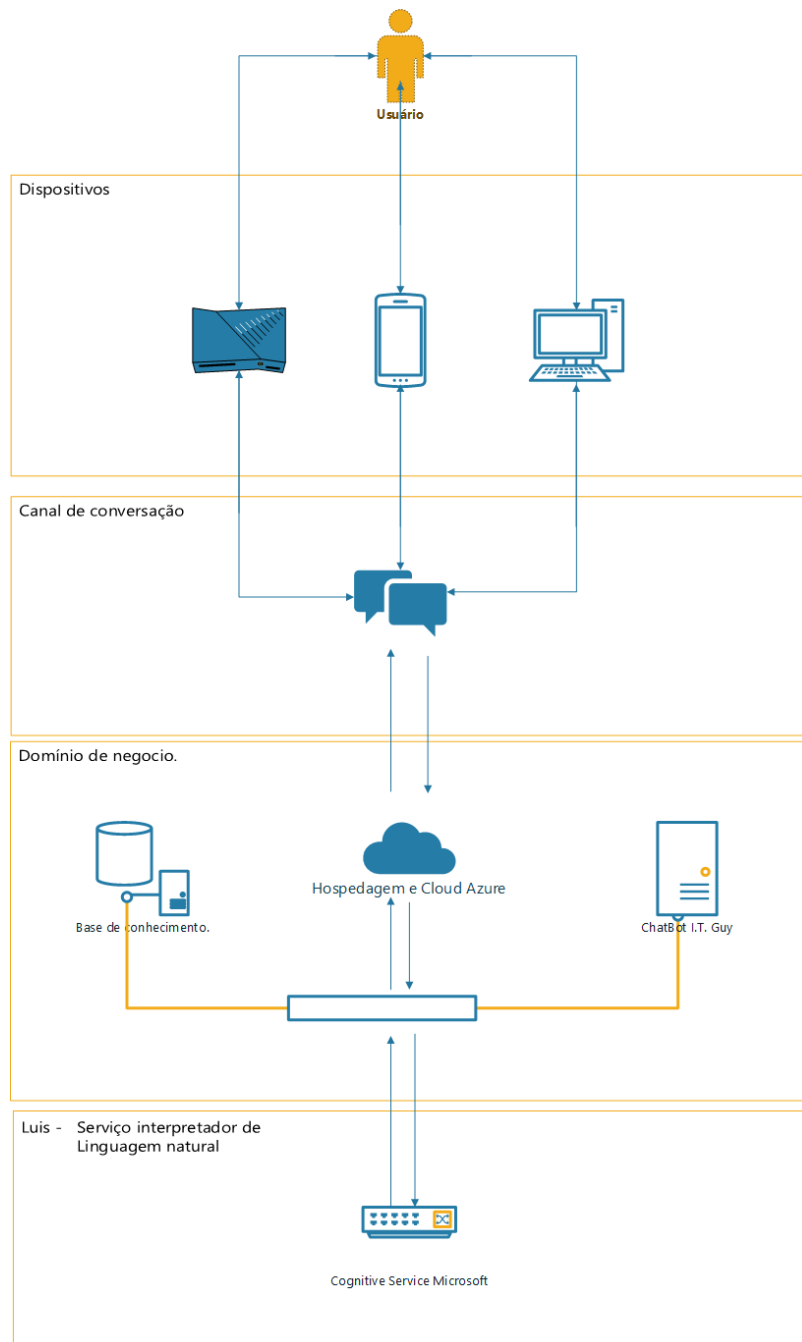
Quanto à sua finalidade trata-se de uma pesquisa aplicada a qual objetiva a geração de conhecimentos para que ocorra aplicação prática, com vistas à solução de problemas específicos. Tem como método dedutivo de abordagem cujos dados foram analisados do macro para o microambiente de forma descendente. Quanto aos procedimentos de coleta de dados trata-se de uma pesquisa bibliográfica que buscou por meio de produções acadêmicas, livros, artigos científicos, não apenas a construção do embasamento teórico, mas também a compreensão do que é o *Service Desk* e um *chatbot* para que o desenvolvimento pudesse ocorrer. Quanto à natureza dos dados, a pesquisa tem abordagem qualitativa (LAKATOS e MARCONI, 2009).

#### 4 DESENVOLVIMENTO

Para criar uma ferramenta que automatiza o atendimento de um *Service Desk* de primeiro nível, precisa-se previamente compreender e entender o que é e como atua a área de *Service Desk* dentro de uma organização. Após o entendimento geral da área pôde-se classificar os níveis de atendimento afim de compreender o que deve ser solucionado no primeiro nível.

Apreendidos os conceitos de uma estrutura de *Service Desk* e a análise técnica dos componentes de *software* necessários para a criação de uma aplicação de *chatbot* desenvolveu-se a aplicação denominada de **ITGuy**, um *software* baseado em PLN que deverá apoiar usuários na resolução de problemas caracterizados como primeiro nível, em uma estrutura de *Service Desk*. A estrutura da aplicação é composta basicamente por três componentes principais, conforme mostra a Figura 1: o canal de comunicação que é por onde o usuário troca mensagens com a aplicação, podendo ser qualquer mensageiro; o interpretador de linguagem natural, que processa a linguagem humana utilizada pelo usuário e identifica os componentes chave e por fim uma aplicação *web* que é responsável por integrar o canal de comunicação e o interpretador de linguagem natural. Cada componente dessa estrutura será melhor detalhado a seguir.

Figura 1 – Arquitetura da aplicação.



Fonte: Próprios autores (2018)

Viu-se que o PLN torna uma máquina capaz de compreender tanto a fala quanto a escrita a fim de convertê-la a uma linguagem inteligível ao computador. Viu-se ainda que, os sistemas de PLN identificam padrões estruturais nas frases a partir de três pilares da comunicação, isto é, a partir do som, da estrutura e do significado que correspondem

respectivamente à fonologia, morfologia e sintaxe, e, semântica e pragmática (PEREIRA, 2012).

Existem diversas tecnologias que são capazes de fazer essa conversão, por exemplo o *Watson* da IBM ou a *Siri* da *Apple*; entretanto o projeto utiliza uma *Application Programming Interface* (API) da *Microsoft* chamada *Language Understand Intelligence Service* (LUIS), que terá importante papel na análise sintática durante a troca de mensagens com os usuários. O LUIS identifica palavras chave na estrutura das frases, como verbos, e a partir disso estrutura os outros elementos gramaticais. Dessa forma, ao se substituir o substantivo *computador* para *notebook* uma *Application Programming Interface* (API) ainda será capaz de entender a frase, pois de acordo com a estrutura frasal, já foi identificado que antes de um verbo há um substantivo e tem por principal função a análise sintática durante a troca de mensagens com os usuários.

O LUIS é utilizado como interpretador de linguagem natural, promovendo assim a tradução das mensagens do usuário para o tratamento das ações na aplicação, este, sendo uma API disponibilizada pela *Microsoft* com uma interface amigável e de fácil usabilidade na integração com os demais elementos do sistema. Sua função é basicamente traduzir a linguagem natural fornecida pelo usuário: interpretar intenções que o usuário deseja executar, como por exemplo: reservar um voo, pagar uma conta ou encontrar um artigo de notícias. Essa sentença pode ser, por exemplo, "Reserve uma passagem para Paris", ou apenas um trecho da frase, como "Reserva" ou "Voo de Paris".

O LUIS trabalha sobre três pilares, sendo estes as intenções, os enunciados e as entidades. As intenções são como o LUIS determina o que um usuário deseja fazer. No caso acima, onde o usuário se interessa em passagens para Paris, suponha-se que o aplicativo esteja sendo utilizado em uma agência de viagens, algumas das intenções a serem cadastradas poderiam ser "Comprar Passagens aéreas", "Reservar voo", "Horários de voos para Recife" e "Viajar para Nordeste", para identificar com assertividade quando seus usuários desejam executar essas diferentes tarefas.

O enunciado é a entrada textual que o LUIS interpretará. LUIS usa primeiro expressões de exemplo que se adicionam a uma intenção, para ensinar a si mesmo como avaliar a variedade de enunciações que os usuários irão inserir.



Uma entidade é usada como uma variável em álgebra, ela captura e passa informações importantes para o *Chatbot*. No enunciado, "Eu quero comprar uma passagem para São Paulo", caso se quisesse capturar o nome da cidade "São Paulo" para uma entidade, como *cidade\_Destino*, seria cadastrada esta relação e após isso LUIS veria o enunciado como "Quero comprar uma passagem para {*cidade\_Destino*}". Essas informações agora podem ser passadas para o *Chatbot* e usadas para concluir uma tarefa.

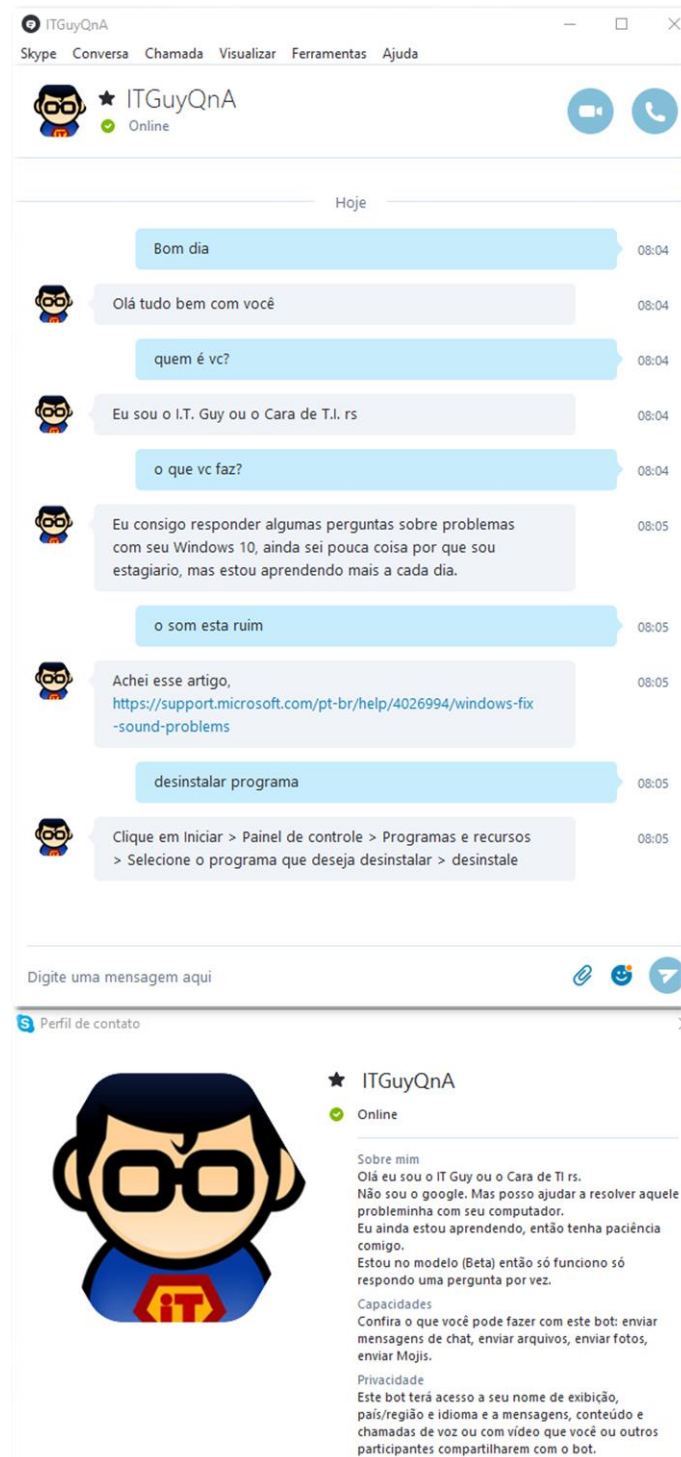
A *WebAPI* foi desenvolvida em Linguagem C#, *FrameWork* 4.6.2, estrutura de projeto MVC (*Model - View - Controller*), o objetivo desse componente do *software* é o de integrar o *Language Understand Intelligence Service* (LUIS) com o canal de comunicação, após o usuário interagir, o LUIS captura a requisição enviada através do canal de comunicação, tem a tarefa de traduzir a linguagem natural do usuário (Português) a uma porcentagem indicando a possível intenção do usuário, a *WebApi* recebe essa intenção e após o processamento da requisição envia solução ou a continuação da interação ao usuário.

A linguagem de programação C# foi escolhida, pois há um grande suporte da comunidade de desenvolvimento de *software*, sendo uma linguagem madura e robusta para criação das mais diversas soluções, além de os integrantes do time de desenvolvimento possuírem mais familiaridade com os recursos presentes nesta linguagem como *frameworks* e *Integrated Development Environment* (IDEs).

Para o desenvolvimento do *chatbot* foi necessária a utilização de um *template* de projeto denominado *BotFramework*, que fornece um ambiente integrado desenvolvido especificamente para o desenvolvimento de *bots*, permitindo que se construa, conecte, teste, implemente e gerencie *bots* inteligentes, tudo em um só lugar. A razão pela qual definiu-se o *BotFramework* da *Microsoft* como principal biblioteca de desenvolvimento, é devido a sua facilidade de implementação na integração utilizando um único código, para diversos canais de conversação, tais como: *Slack*, *Telegram*, *Messenger*, *Skype*, *Whatsapp*, entre outros.

Para a publicação de serviço optou-se por um serviço modelo *Plataform as a Service* (PAAS) oferecido pela *Microsoft* através do *AZURE*, sendo esta uma das maiores plataformas de *cloud computing* disponíveis atualmente, com resiliência e confiabilidade, além de oferecer maior interação com as tecnologias *Microsoft*. A Figura 2 apresenta a tela de interação do Usuário com o **ITGuy**.

Figura 2 – Canal de Conversação Skype



Fonte: Próprios autores (2018)

O LUIS utiliza modelos matemáticos de análise para determinar qual é a ação mais provável para determinada intenção do usuário, ou seja para uma frase como: “voou para russia” ou como “passagem para russia”, o algoritmo, se assim treinado, poderia responder como 70% de probabilidade que as duas frases se tratam de uma reserva de passagem área mesmo que o usuário use palavras diferentes a aplicação interpreta a intenção da frase e então divide os 30% restantes para as probabilidades que mais se encaixem em outras intenções. Após esta análise repassa estas informações para o *chatbot* que tomará a decisão com base na sua programação e base de conhecimento e responderá para o usuário através do canal de conversação.

Este modelo deve ser retroalimentado para que a acurácia seja satisfatória, e com base no funcionamento de *Service Desk* exposto anteriormente, esta atividade de retroalimentação pode ser feita pelo atendimento de segundo ou terceiro níveis, pois caso uma resposta não seja interpretada esta ficará em uma base de dúvidas para que a equipe responsável pelo treinamento do *bot*, diga ao LUIS qual era a intenção por trás dos dizeres do usuário e qual a ação que deverá ser tomada.

Quanto mais intenções são cadastradas maior será o nível de assertividade, até mesmo para modelos não cadastrados, como por exemplo, caso sejam cadastrados “Oi” e “Olá” como intenções de saudação e a ação seja uma resposta de saudação, se o Usuário digitar “Bom dia”, o algoritmo entenderá que pode se tratar de uma saudação e responderá com a probabilidade desta intenção.

## 5 RESULTADOS E DISCUSSÃO

O **ITGuy** foi concebido na forma de prova de conceito (POC - *Proof Of Concept*), onde é desenvolvido um produto mínimo viável (MYP- *Minimum Viable Product*) para se validar a se proposta de solução é atendida pelo produto desenvolvido. Dentro do cenário proposto o *Chatbot* mostrou-se eficiente na resolução de requisições menos complexas, pois uma vez que sua base de conhecimento seja alimentada com as soluções para os incidentes mais recorrentes dentro do *Service Desk*, as demandas que remetem a esses problemas cadastrados são solucionadas em um tempo muito menor em comparação ao fluxo de atendimento comum, que seria através do atendimento humano.

Se considerado que as demandas mais simples serão atendidas pelo *bot*, a equipe técnica poderá se dedicar em alimentar a base de conhecimento e atuar nas demandas de maior complexidade, isto é, as do segundo e do terceiro nível, o que permite uma maior especialização da equipe de atendimento bem como a melhoria na agilidade dos atendimentos mais simples, o que é benéfico para a empresa pois os usuários solicitantes ficam cada vez menos tempo em espera para resolução de seus problemas e, nos casos mais complexos, a solicitação é atendida com uma qualidade maior pelos níveis acima.

## 6 CONSIDERAÇÕES FINAIS

O objetivo geral foi alcançado e os problemas de pesquisas foram respondidos, pois concluiu-se o desenvolvimento de uma aplicação de *Chatbot* capaz de auxiliar e resolver problemas de um *Service Desk* para primeiro nível, agilizando-se o atendimento de requisições menos complexas e direcionando as demais requisições para os analistas responsáveis dos níveis acima por resolve-las.

Verificou-se que durante o processo de desenvolvimento e teste da aplicação que a maioria das requisições feitas ao sistema a resposta é satisfatória. Contudo, existe uma chance de o sistema não conhecer uma possível solução para determinado caso ou mal interpretar a intenção do usuário. Para esses casos, deve-se haver uma equipe especializada em desenvolver e melhorar a base de conhecimento da aplicação e retroalimentá-la com novas ações, para que nas próximas interações o sistema seja capaz de resolver a requisição.

Assim, sugere-se que futuros trabalhos deem continuidade às pesquisas com vistas a alimentação do sistema com um número maior de intenções e entidades condizentes a demais necessidades dos usuários.

## REFERÊNCIAS

CAVALARI, G. O. T., COSTA, H. A. X. **Modelagem e Desenvolvimento de um Sistema Help-Desk para a Prefeitura Municipal de Lavras – MG.** RESI - Revista Eletrônica de Sistemas de Informação. v. 4, n. 2 >. Disponível em: <<http://www.periodicosibepes.org.br/index.php/reinfo/article/view/158/50>>. Acesso em: 19 mar. 2018.

CHANTAROTWONG, B. “**The learning chatbot,**” Final Year Project, 2005.

COHEN, R. **Implantação de Help Desk e Service Desk.** São Paulo: Novatec Editora Ltda, 2008.

FERREIRA, L. UCHÔA, J. **Desenvolvimento de um chatbot para auxiliar o ensino de espanhol como língua estrangeira,** Revista Bazar, 2005. Disponível em: <[http://www.educadores.diaadia.pr.gov.br/arquivos/File/marco2012/espanhol\\_artigos/leandro\\_ferreira\\_joaquimuchoa.pdf](http://www.educadores.diaadia.pr.gov.br/arquivos/File/marco2012/espanhol_artigos/leandro_ferreira_joaquimuchoa.pdf)>. Acesso em: 27 mai. 2018.

LAKATOS, E. M.; MARCONI, M. A. **Fundamentos da Metodologia Científica.** 6 ed. São Paulo: Atlas, 2009

MEDEIROS, L. C. L. L. SOARES, W. **Formação de Suporte Técnico.** Rio de Janeiro: Rede Nacional de Ensino e Pesquisa, 2010.

PEREIRA, S. **Processamento de Linguagem Natural.** Departamento de Tecnologia da Informação da Faculdade de Tecnologia de São Paulo. 2012. Disponível em: <<https://www.ime.usp.br/~slago/IA-pln.pdf>>. Acesso em 25 mar. 2018.

RUSSEL, S. NORVIG, P. **Artificial Intelligence. A Modern Approach.** 3. Ed. New Jersey: Pearson Education Inc, 1995.

SOTTO, Léo., COSTA, Victor. **Chat Bot com aprendizado a partir de diálogos.** Instituto de computação da Universidade Estadual de Campinas. Disponível em [http://www.ic.unicamp.br/~ffaria/ia1s2015/projects/Chatbot\\_Leo\\_Victor.pdf](http://www.ic.unicamp.br/~ffaria/ia1s2015/projects/Chatbot_Leo_Victor.pdf). Acessado em 24 MAR 2018.

**“O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es).”**