MARINABOT: CONSTRUÇÃO DE UM SIMULADOR DE DIÁLOGO PARA ASSISTÊNCIA ACADÊMICA

Tiago Domingos Cipriano[1]

Jaqueline Brigladori Pugliesi[2]

Resumo

São notáveis as evoluções atingidas pela inteligência artificial, assim como os resultados apresentados pelas indústrias de tecnologia na forma de novos produtos e serviços. E entre os produtos gerados encontram-se os *chatterbot,* um programa que fornece interação por meio de linguagem natural. Este artigo tem como objetivo descrever a construção de um robô de conversação (c*hatterbot*), que opera como um assistente acadêmico para apoiar alunos de uma faculdade na obtenção de informações relacionadas às suas atividades dentro da instituição, exemplificando a utilização de ferramentas, técnicas e recursos que internamente implementam PLN (Processamento de Linguagem Natural). Para que fosse possível o entendimento sobre como projetar tal solução, foram realizadas pesquisas em livros e artigos científicos; teste com bibliotecas, serviços e ferramentas; cursos sobre *chatterbot*; discussões técnicas com programadores, arquitetos de software e outros profissionais de tecnologia.. A tecnologia de *chatterbot* demonstra-se capaz de ser facilitador para o dia a dia dos alunos uma vez que permite o uso de linguagem natural para acesso a informações.

Palavras-chave: *Chatterbot*. Inteligência Artificial. Processamento de Linguagem Natural.

Abstract

It is remarkable the evolution achieved by artificial intelligence, as well as the results shown by the technology industries in the form of new products and services. Among the generated products we have chatterbot, a program that provides interaction through natural language. This article aims to describe the construction of a chat robot (chatterbot), which operates as an academic assistant to support students of a college to obtain information related to activities inside the institution, exemplifying the use of tools, techniques and resources which internally implement PLN (Processing of Natural Language). In order to be possible the understanding of how to project such a solution, some research in books and scientific articles was done; library, service and tool tests were performed; chatterbot courses and technical discussions with programmers, software architects, and other technology professionals were done. The chatterbot technology demonstrates itself to be a daily facilitator for the students as it allows the use of natural language for access to information.

1 Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: tiagoandroidti@gmail.com.
2 Doutora em Ciências da Computação pela USP – São Carlos/SP. Endereço eletrônico: jbpugliesi@gmail.com.

Keywords: *Artificial Intelligence. Chatterbot. Natural Language Processing.*

1 Introdução

Com a expansão da internet e a propagação em larga escala da utilização de dispositivos móveis conectados à rede, habilita-se a ocorrência de infinitas possibilidades de interações entre usuários. Como exemplo, têm-se as redes sociais *on-line* de relacionamentos, em que a interação entre os mesmos ocorre pela troca de mensagens de texto, multimídia e vídeos.

Neste cenário, o usuário é cada vez mais consumidor de informações e serviços de instituições, empresas ou corporações, entre outros. Desse modo, surgem novas necessidades e por meio do seu dispositivo o usuário não só quer se comunicar com um amigo ou parente, mas também ter acesso a informações e resolver problemas do dia a dia, dos mais variados ambientes ao qual está inserido, utilizando o seu equipamento conectado à *Internet*.

Algumas empresas disponibilizam *call centers* para suprir as necessidades dos seus clientes, assim como instituições de ensino criam novos meios de propagação de informações para os seus diversos públicos. Porém, como as demandas estão crescendo exponencialmente, torna-se impossível a contratação exponencial de novos atendentes supri-las.

Assim, novos mecanismos computacionais surgem para auxiliar o atendente e automatizar o processo de interação entre o usuário e as corporações, ou seja, a interação não é mais apenas entre o usuário e um atendente e sim também com um computador.

Chatterbot é um programa que fornece uma interação por meio de linguagem natural, isto é, faz uso da inteligência artificial para permitir que o usuário atinja seus objetivos, sendo que a abrangência do robô limita-se respeitando o propósito para o qual o *bot* foi projetado para atender.

A utilização de *Chatterbot* não é limitada somente a empresas e se expande por universidades, hospitais e etc. Silva (2011) apresenta a utilização do mesmo em várias outras situações como: sistema de estudo terapêutico, atendimentos *on-line*, para auxílio nas dúvidas dos alunos em ambientes educacionais ou ainda como sistema de entretenimento. Entre os vários exemplos de missões que são atribuídas aos *bots* encontram-se: a captura de dados, assistentes de voz e o *FAQbot*.

Este artigo tem como objetivo descrever a construção de um robô de conversação (*chatterbot*), que opera como um assistente acadêmico para apoiar alunos de uma faculdade na obtenção de informações relacionadas às suas atividades dentro da instituição, exemplificando a utilização de ferramentas, técnicas e recursos que internamente implementam PLN (Processamento de Linguagem Natural).

Neste trabalho, apenas a título de exemplificação, foi considerado que o *chatterbot* desenvolvido foi para atender aos alunos da Fatec Franca, porém a arquitetura do projeto suporta adaptações, permitindo o uso em outras instituições de ensino.

O programa MarinaBot, permite que a interação entre os usuários e o computador ocorra por textos inseridos em uma interface gráfica *web* ou por voz onde a solicitação falada pelo usuário é recebida por um programa local escrito em *python*, e o robô interpreta as mensagens e define possíveis respostas ao usuário. Esta interação, entre usuário e computador realizada em linguagem natural chama-se *Chatterbot.*

2 Referencial Teórico

Os estudos direcionados sobre as possibilidades de construir computadores com a capacidade de compreensão e simulação de comportamentos humanos têm como marco histórico o jogo da imitação proposto por Alan Turing (por volta de 1950) que veio posteriormente a ser conhecido como Teste de Turing. Entre os vários aspectos da definição do que é Inteligência Artificial, Russel e Norvig (2013) descrevem que uma abordagem centrada nos seres humanos deve, ser em partes, uma ciência empírica envolvendo hipóteses e confirmação experimental. Nesta linha, um teste foi proposto por Turing em que o sistema passará no teste se um interrogador humano, depois de propor algumas perguntas por escrito, não consiga descobrir se as respostas escritas foram providas por uma pessoa ou por um computador. Neste contexto, há um item em comum entre as capacidades das quais tantos os robôs de conversação quanto o computador necessitam para realizar o teste de Turing*:* a implementação de mecanismo que permita o uso de processamento de linguagem natural, uma vez que os habilita para que se comuniquem em um idioma natural com os seres humanos (RUSSEL; NORVIG,

2013).

Uma vez que a máquina tem a habilidade de se comunicar como um humano, Laven descreve o *chatterbot* como um programa, que faz o uso de Inteligência Artificial, que tenta simular uma conversa digitada, com o objetivo de enganar temporariamente um humano, de modo que este pense que estava conversando com outra pessoa. Para que isso seja possível, o PLN se apoia em outras áreas do conhecimento, tais como matemática, linguística, psicologia e estatística.

Ferrucci et al (2010) afirma que os *chatterbots* são sistemas atrativos e extremamente desafiadores para a área de Computação e Inteligência Artificial, nos quais envolvem-se a recuperação de informações, processamento de linguagem natural, representação de conhecimento, aprendizado de máquina e interfaces humano-computador. Alguns desses desafios foram enfrentados pelo *chatterbot* Eliza na década de 60.

Eliza foi um *chatterbot* construído em 1966, no MIT (*Massachusetts Institute of Technology*), pelo professor Joseph Weizenbaum, e é um robô que simula um psicanalista em uma conversa com um paciente, porém, compreende-se como alguns dos seus problemas, a incapacidade de lembrar-se do que já havia dito ao usuário e também o fato que o robô responde ao interlocutor usando partes das frases que o paciente usou ao conversar com o *bot*, gerando às vezes diálogos um tanto confusos (LEONHARDT et al, 2003).

No cenário brasileiro há vários casos de soluções onde a inteligência artificial tem ajudado empresas em várias demandas, como por exemplo o BIA (Bradesco Inteligência Artificial) que tinha como desafio otimizar o atendimento telefônico sobre produtos e serviços com clientes e funcionários das mais de 5 mil agências de todo o Brasil, a plataforma de desenvolvimento do BIA é o IBM Wastson (GRAÇA, 2018).

Andrade (2012) cita o caso de sucesso da Professora Elektra, um *chatterbot* brasileiro destinado ao ensino de conceitos sobre redes de computadores. Elektra é um dos primeiros *chatbots* educacionais que se tem conhecimento no Brasil. Thaise (2016) descreve no site Medium, que o projeto foi desenvolvido por um grupo de pesquisa da Federal do Rio Grande do Sul (UFRGS), baseado no software gratuito ALICE, sendo colocada à disposição dos internautas em meados de 2002. Os desenvolvedores do *chatterbot* Professora Elektra relatam que a ferramenta pode ser aprimorada pois ao catalogar, por meio dos *logs*, as perguntas feitas sobre os tópicos ao qual o robô atendia, notaram que perguntas realizadas por pessoas de

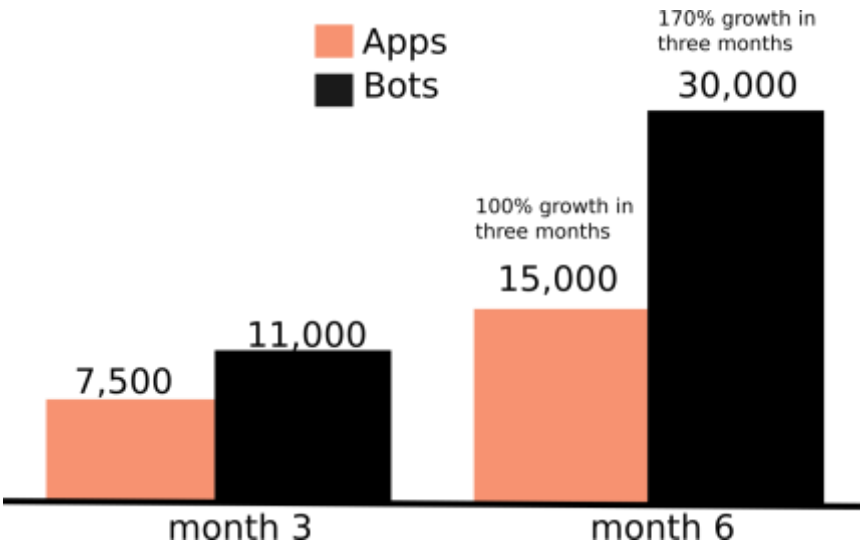
regiões geograficamente diferentes do Brasil geravam incompreensão por parte do *bot*.

O site Chatbots Magazine é uma das fontes mais lidas e respeitadas de informações sobre *chatbots*, com mais de 100.000 leitores e 250.000 leituras todo mês. Neste site, foi publicado em março de 2017, pela equipe de desenvolvimento do BRAIN (plataforma que fornece uma suíte de gerenciamento para *chatterbot* e *blockchain*) da empresa alemã, TechFunder UG, o artigo *online* Relatório Chatterbot 2018: Tendências Globais e Análises, que contém dados relevantes sobre o estado atual dos *chatterbots* (CHATEBOTS MAGAZINE).

O artigo trata da aceitação de grandes *players* de tecnologia em relação aos *chatterbots*, por empresas como Microsoft, Facebook, Google, Amazon, IBM, Apple e Samsung que trabalharam para criar seu próprio *chatbot*; criaram plataformas abertas e interfaces para que a aceitação do *chatbot* pela sociedade passe mais fácil e rapidamente. O texto também descreve a criação de alguns produtos relacionados ao tema que surgiram no período entre 2010 e 2016: em 2010 surgiu a Siri, assistente pessoal para iOS, macOS e watchOS exclusivo da Apple; o Watson começou em 2011 (plataforma de serviços cognitivos da IBM para negócios); a versão piloto do assistente de voz Samsung Bixby apareceu em *smartphones* em 2012; a Alexa, assistente virtual inteligente da Amazon, aprendendo a responder perguntas desde 2014; Durante o desenvolvimento deste artigo, ao ‘conversar’ com o *chatterbot* da BRAIN, foi obtido a informação de que a empresa que o criou, iniciou suas atividades neste seguimento no ano de 2015 e o Assistente do Google (Ok Google) ganhou sua forma moderna em 2016. Dados divulgados pelo Facebook por meio do gráfico descrito na Figura 1 mostram a relevância que os *bots* têm alcançado em sua plataforma quando comparada ao crescimento destes em relação ao crescimento de apps (CHATTERBOT REPORT 2018: GLOBAL TRENDS AND ANALYSIS, 2018).

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

Figura 1 – *Initial Grow of Apps Vs. Messenger Bots.*



Fonte: Chatbot Report 2018: Global Trends and Analysis, 2017, online.

Como cada consumidor tem suas necessidades exclusivas e a maioria dos aplicativos e soluções de negócios disponibilizados pelas plataformas de serviços são altamente personalizáveis, os esforços práticos para facilitar o uso de recursos em nuvem no formato *Software as a Service* (SaaS), apresentam-se muitas vezes, por meio da disponibilização de recursos como *Software Development Kit (SDK)* ou *Application Programming Interface Representational State Transfer (API Rest).* Tem-se assim os SDK que, em geral, oferecem uma maneira programática de acessar dados e/ou configurações nos aplicativos, tais dados são representados como atributos das mais diversas entidades. Os SDKs precisam ser fáceis de usar para permitir que um fornecedor de *software* independente ou outros usuários desenvolvam aplicativos avançados sobre os aplicativos originais, para isso um SDK pode ter uma interface de programação de aplicativo (API) que permitirá que um desenvolvedor atualize programaticamente um atributo de código postal de uma conta por exemplo (MICROSOFT CORPORATION, 2006).

API é o acrônimo de *Application Programming Interface* ou, em português, Interface de Programação de Aplicativos, em que a tecnologia de implementação tem menor relevância, pois o seu real valor é gerado pela capacidade que pode fornecer aos aplicativos e processos das plataformas de negócios digitais. Atualmente, é o padrão pelos quais as empresas trocam dados e constroem experiências consistentes de clientes entre canais, onde a conexão entre processos

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

de negócio, serviços, conteúdos e dados para parceiros de canal, equipes internas e desenvolvedores se manifestam de maneira fácil e segura. A Internet das Coisas também conhecida como IoT (*Internet of Things*) ou IoE (*Internet of Everything*), refere-se a objetos comuns, reinventados como dispositivos programáveis, conectados à Internet, e fazem uso constantes das *WebApis*, como os relógios inteligentes que exibem notificações de atualizações do mercado de ações ou de esportes. As *WebApis* são escritas usando, entre outros, o protocolo *Representational State Transfer* (*REST*) que se consolida como o padrão mais adotado na atualidade por ser um estilo arquitetônico em vez de um padrão rigoroso, logo, sua flexibilidade e liberdade de estrutura sugere um guia de melhores práticas de design (MULLOY, 2012; NIJIM, PAGANO, 2014).

Entre os vários seguimentos de desenvolvimento, o foco em soluções otimizadas para dispositivos moveis cresce a cada dia, trazendo novos conceitos e abordagens, entre elas encontram as aplicações progressivas, que usam as *API Rest* como parte de sua premissa de arquitetura.

Pontes (2018) explica que PWA é um acrônimo para *Progressive Web Apps* ou, em português, Aplicações Web Progressivas, e o foco deste conceito é a melhoria contínua focada em atender o maior número de pessoas possíveis, e trata-se da criação de uma experiência a acesso contínuo sem nenhum tipo de restrição tecnológica. Assim sendo, usuários de microcomputadores, *tablets*, *smartphones*, com *Browser* atualizados ou obsoletos, com ou sem *internet* podem continuar a usar a solução ao longo de suas novas versões. Para que de fato um app seja um PWA, uma série de requisitos são necessários:

- Registrar um *Service Worker* conforme especificado na *W3C*.
- Resposta com *status* 200, mesmo estando em *off-line*.
- Exibir conteúdo quando o *JavaScript* estiver desabilitado.
- O uso do protocolo HTTPS (*Hyper Text Transfer Protocol Secure*) é obrigatório na comunicação entre *frontend* e *backend*.
- Deve carregar rapidamente no 3G.
- Deve questionar o usuário se deseja instalar a aplicação.
- Deve ser configurada com uma *splash screen* personalizada.

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

O conceito de desenvolvimento de software, onde a abordagem inicial e principal são os dispositivos móveis, tem o nome de *mobile first.* Esta forma de trabalhar propõe a criação da organização das páginas com o foco nos dispositivos *mobile*, mas também levando em consideração os usuários de microcomputador (PONTES, 2018).

Web scraping ou “raspagem de dados” é a prática de copiar ou transformar páginas da *Web* em bases de dados, usando aplicativos específicos. Com isso, é possível automatizar o processo de coleta de dados em *websites* que não os colocam à disposição em formatos diretamente manejáveis, como CSV (*Comma Separated Values*), XML (*Extensible Markup Language*) ou TXT (arquivo de texto simples). Os dados a serem raspados podem estar integrados a um documento HTML (*Hypertext Markup Language*) em português Linguagem de Marcação de Hipertexto, ou em formatos ilegíveis por aplicativos de tratamento de dados, como DOC (arquivo de texto com estruturas mais elaboradas do que o txt) ou PDF (*Portable Document Format*), ou ainda distribuídos em diversas páginas (SANTOS, 2015).

Por sua vez, *WebCrawler* é um software, também chamado de motor de busca, que se resume a procura de dados em *website* guardando-os numa base de dados, outra forma de implementação dessa ferramenta descreve-se por uma *Query* direta aos *Websites* relevantes procurando pelos dados necessários (TRÄSEL, 2017). Neste segundo método, obtém-se os dados atualizados, pois o acesso é feito quando o motor recebe um pedido pelo utilizador, ocasionando um certo tempo para processamento e ainda dependendo da disponibilidade do site no momento da solicitação (RODRIGUES, 2016).

3 Desenvolvimento

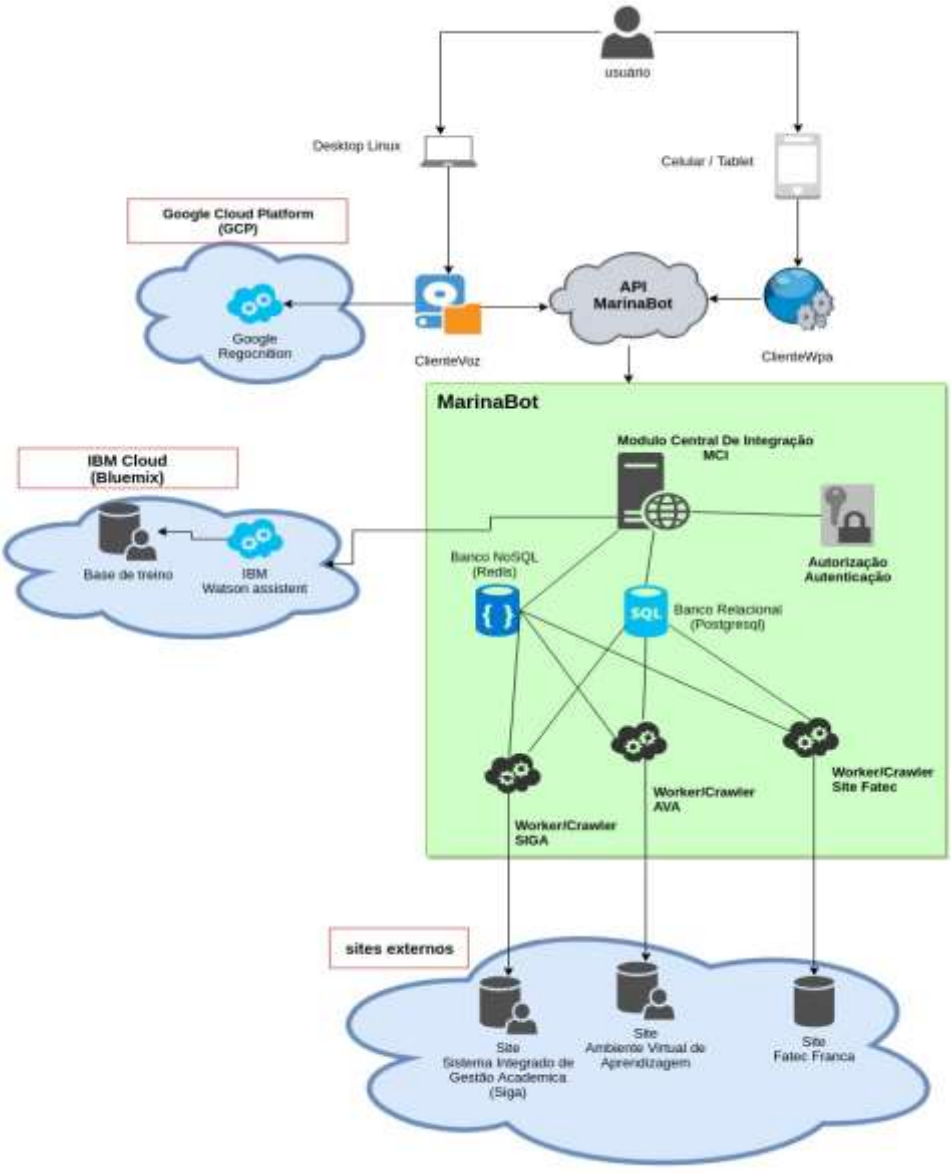
Para que fosse possível o entendimento sobre como projetar tal solução, foram realizadas pesquisas em livros e artigos científicos; teste com bibliotecas, serviços e ferramentas; cursos sobre chatterbot; discussões técnicas com programadores, arquitetos de software e outros profissionais de tecnologia.

O projeto MarinaBot é uma maneira de permitir que os alunos se comunique com computador de uma forma mais humana. A interface gráfica permite que os usuários possam ter uma iteração textual com o *bot* por meio da internet, podendo o

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

dispositivo ser um *tablet*, *smartphone*, ou qualquer outro que possua um navegador. As W*ebApis* expostas pelo módulo central de integrações permitem que os processos de negócio ou a exposição de dados possam ser flexíveis, e o uso deste recurso permite que várias modalidades de clientes possam ser desenvolvidas para representar a mesma solução, ou seja, novas interfaces de comunicação com os usuários podem ser conectadas a aplicação principal. O projeto ao qual este artigo se refere, em alto grau de abstração, encontra-se configurado na arquitetura representada na Figura 2.

Figura 2 – Arquitetura do projeto MarinaBot.



Fonte: Autor.

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

3.1 Interface do usuário

Os módulos ClienteVoz e ClienteWpa são dois programas distintos que consomem informações de uma mesma fonte de dados, porém, a representação da interação com os usuários fornece experiências completamente diferente entre si.

O Módulo ClienteWpa é uma solução de acesso em que o usuário pode interagir com o robô de conversação por meios de uma interface gráfica utilizando o seu celular, *tablet* ou qualquer outro dispositivo em que possa ter acesso a um navegador *web*. Esta solução se trata de um *frontend* que implementa os conceitos e princípios do PWA e *mobile first*, permitindo que a aplicação possa ser utilizada pelo maior número de alunos.

A segunda forma de interação é um cliente que, por meio de *scripts* escrito na linguagem de programação Python, permite acesso por voz na comunicação com o robô. No entanto, para que seja possível a comunicação deste módulo com o Módulo Central de Integração (MCI) é necessário que os comandos informados pelo usuário estejam em formato de texto e para isso é necessário uma conversão que pode ser obtida através do uso de serviços fornecido por terceiros, como é o caso do *Speech To Text* (STT) oferecido pela Google, onde, por meios de *apis,* é enviado o áudio contendo a fala do usuário coletado pelo módulo de voz. Obtido um texto que representa o conteúdo do som enviado. Tal retorno é enviado para o MCI que, após processar as informações recebidas, retorna um objeto do tipo JSON (*JavaScript Object Notation*) para o módulo de voz. Este objeto possui um atributo que contém um texto com a resposta que será apresentada ao usuário do MarinaBot e, uma vez que este processo acontece, é necessária uma nova conversão, mas desta vez de texto para áudio. Novamente o projeto optou por recorrer a serviços externos, desta vez utilizando o *Text To Spech* (TTS) da Google e como resposta se obtém um áudio que é reproduzido para o usuário. A Figura 3 demonstra uma interação real do usuário com o Marinabot:

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

Figura 3 – Conversa entre usuário e *bot*

Fonte: Autor.

3.2 Integração com Watson Assistent

Mesmo os programadores que não possuem conhecimento profundo nos campos teóricos de Inteligência Artificial, tem a possibilidade de construir projetos que fazem uso de seus recursos. Entre as ferramentas cognitivas da plataforma *IBM Cloud*, encontra-se o Watson Assistent (antigamente conhecido como Watson Conversation) que oferece os recursos necessários para o desenvolvimento de *chatterbots*. Após realizar as etapas de cadastro e criação do recurso de conversação, é necessário criar um *workspace* (uma organização de todos os elementos necessários para que o *bot* funcione, tais como intenções, exemplos de perguntas dos usuários, entidades e demais itens), assim como a configuração no idioma ao qual o robô se comunicará com o usuário final.

O Watson, a princípio, não deixa de ser uma máquina, um programa que o desenvolvedor precisa configurar suas entradas que são exemplos de perguntas e texto que o *bot* receberá; para que o mesmo possa entender o que deve ser respondido (as respostas representam as saídas do programa pós-processamento), esse conjunto de entradas são obrigatórios, pois são os dados de treinamento para

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

do *bot*. Levando-se em consideração os objetivos do *chatterbot*, o público-alvo e o contexto que o mesmo se encontra, quanto mais dados no formato de perguntas houverem, e quanto mais reais elas forem, melhor será a assertividade do robô em entender o que o usuário deseja e procura, ou seja, as classificações das intenções tendem a ser mais precisas. Dentro dos conceitos do Watson há um elemento denominado intenção (*Intents*), que representa um objetivo ou tema alvo que o desenvolvedor usa para planejar quais são e como são caracterizados tópicos conhecidos pelo *bot*, tais tópicos são utilizados como padrões a serem detectados na conversação.

A documentação do Watson define intenção como propósitos ou objetivos expressos na entrada de um cliente, tais como responder a uma pergunta ou processar um pagamento de fatura. Em termos práticos, se é desejado que o *bot* atenda assuntos cujo o tópico principal seja qual a composição do corpo de docentes da instituição, é necessário adicionar uma nova intenção na ferramenta, cujo título pode-se assumir como o seguinte: #cursos_oferecidos_pela_faculdade, e atrelado a essa intenção constam os exemplos de usuário que são possíveis casos de perguntas sobre este tópico (Quem são os professores da instituição?, Qual a composição do corpo docente da unidade de Franca?, Quais são os professores e mestres da faculdade?, Quem faz parte do corpo docente?, etc.). Conforme são adicionadas novas perguntas, o Watson inicializa um processo de treinamento das novas entradas fornecidas.

A Figura 4 exemplifica o resultado final de uma iteração quando uma intenção valida é detectada.

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

Figura 4 – Intenção detectada: Cursos_Oferecidos

Fonte: Autor.

Uma vez que são definimos intenções e fornecidos exemplos para que o bot possa ser treinado, é necessário dar a habilidade para que ele possa responder à pergunta realizada pelo usuário. Este recurso se encontra com o nome de *Dialog*, e possui uma estrutura baseada em pastas, nó e sub nó, fornecendo meios de não só responder ao usuário, mas também fazer novas perguntas, permitindo assim o engajamento da conversa. Cada nó do diálogo possui um nome para que o desenvolvedor tenha a possibilidade de identificar ao que se refere; a concatenação dos operadores *and* e *or* entre entidades e intenções, um conjunto customizável de respostas de texto (há a possibilidade de configurar as respostas como randômicos ou sequenciais) e a opção de fazer uma pergunta e aguardar a resposta do usuário.

Alguns tópicos, por mais específicos que sejam no escopo em que se encontram, abrem margens para um desdobramento de subníveis de informações que o usuário deseja obter. Em uma situação hipotética, dado que a existência de duas intenções do campo de domínio do robô: #calendario-de-provas e #calendario-

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

de-trabalhos, e os seguintes exemplos de perguntas (Há alguma prova para a semana que vem? Existe algum compromisso para o dia 31 de outubro? Quais são as datas das próximas provas?) somente a detecção da intenção do usuário não é o suficiente para compreender em plenitude o que deve ser respondido, pois tais intenções são amplas.

Para tratar esses tipos de cenários a ferramenta disponibiliza um recurso chamado *Entities* (Entidades) que são vinculadas a uma intenção. Com essa funcionalidade, permite-se que marcações de especificidades possam ser nomeadas e identificadas.

Nos exemplos de intenções apresentadas, os momentos em que as provas ou os eventos de calendários vão se realizar, podem ser relevantes aos usuários.

As entidades são marcações de valores textuais, que o bot detectará e informara qual a marcação e qual o valor do conteúdo detectado para o devido tratamento das repostas na etapa de diálogo.

A IBM permite que o desenvolvedor crie as suas próprias entidades, assim como fornece modelos padrões que são comumente utilizadas pelas comunidades de desenvolveres que utilizam a ferramenta evitando assim que a roda seja reinventada.

No painel de desenvolvimento oferecido pela IBM, as entidades são representeadas por @nome-da-entidade e cada uma delas pode possuir vários apelidos, sendo assim, uma entidade chamada @cidade, com o valor: São Paulo, pode assumir como *alias* (usado para indicar que uma entidade também é conhecida ou familiarizada por outro nome) SP ou terra da garoa.

Com isso, pode-se adicionar as condições dos diálogos, gerando novos aspectos de resposta quando as entidades são detectadas nas entradas dos usuários.

3.3 Acessos aos dados

Uma das etapas do desenvolvimento do projeto MarinaBot envolve a aquisição de dados e informações que se encontram em três repositórios externos: o Sistema Integrado de Gestão Acadêmica (SIGA), o Ambiente Virtual de Aprendizagem (AVA) e o site da Fatec Franca, conforme descrito na Figura 2.

Para cada base de informação, existe um mecanismo de extração de dados baseado em *web crawler* e *scraping*.

Realizada a aquisição os esses dados são inseridos em uma base de dados onde o servidor *backend* tem acesso.

No projeto estes recursos são implementados usando duas ferramentas que em conjunto formam um único módulo nomeado no projeto de WorkerCrawler. A primeira delas é um *framework* chamado Scrapy, que é responsável por realizar o acesso aos sites necessários para extrair os dados necessários.

Quando o usuário solicita alguma informação ao *bot* a mesma previamente já está na base de dados de domínio da solução, dessa forma, o retorno do dado ocorre no tempo de retorno do banco. Porém, o acesso às fontes externas não é uma tarefa síncrona.

Para determinar a periodicidade em que as coletas de dados serão realizadas foi utilizado no projeto a ferramenta Celery que fornece um suporte de agendamento de tarefas distribuídas. Cada tarefa é uma unidade de execução.

No projeto tais tarefas são os WorkerCrawler que podem ser executados de forma independentes e são assíncronos.

3.4 Módulo Central de Integração (MCI)

O MCI é o ponto de concatenação entre os módulos clientes, comunicação com o Watson Assistent, regras de autenticação e autorização de acesso dos alunos e regras de integração dos dados com as respostas do *bot*.

Neste processo, sempre que um usuário devidamente autorizado faz uma pergunta ao Marinabot, o MCI monta uma requisição para o Watson.

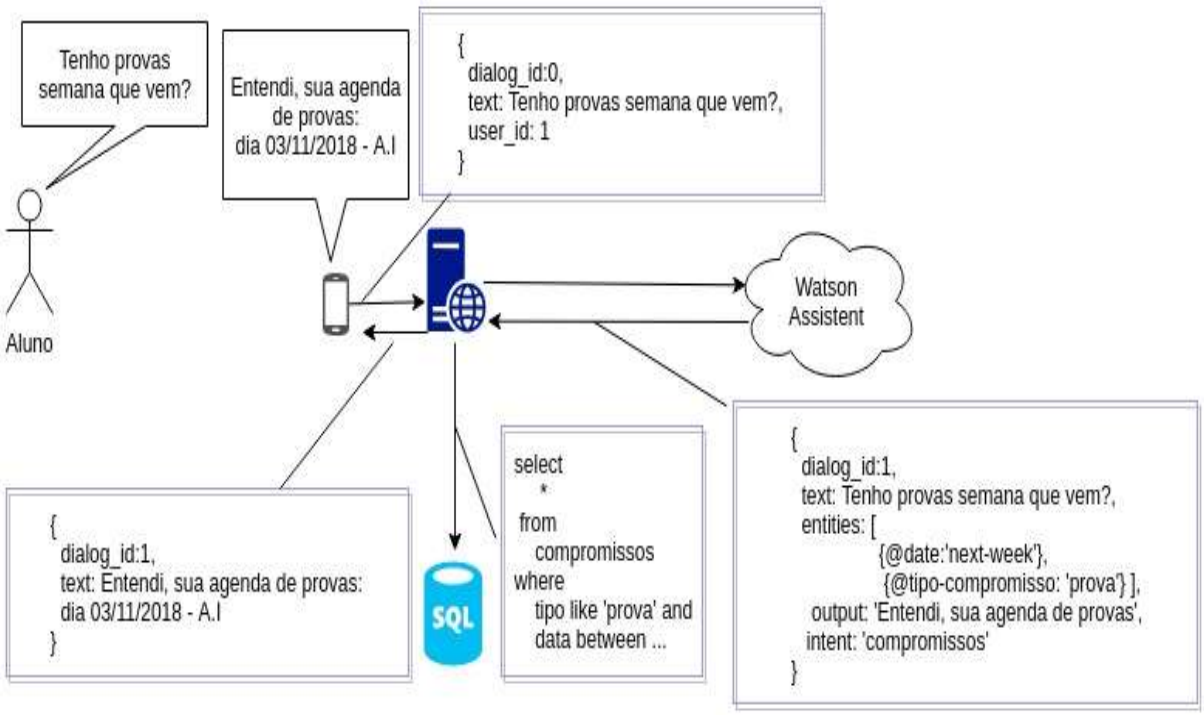
A resposta é um objeto que contém vários atributos, porém nesta etapa são utilizados: o identificador de diálogo (um código único que permite o gerenciamento da continuidade da conversação), a intenção do usuário e as *entidades* demarcadas como encontrada pelo Watson.

Uma vez que se tem acesso às entidades e à intenção, o *backend* faz a interpolação das informações que constam no banco de dados, com a resposta de texto fornecidas pelo Watson.

O retorno oferecido ao cliente solicitante é montado e então enviado para o *frontend*, conforme exibido na Figura 5.

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

Figura 5 – Fluxo de informações da requisição do usuário



Fonte: Autor.

Um segundo fluxo é quando o *bot* precisa colher mais detalhes do usuário para atender uma requisição. Neste contexto, o cliente sempre envia o id de diálogo, repassado pelo MCI. Esse cenário acontece quando mais de uma entrada de dados é necessária ou quando o *bot* não entende o que o usuário solicita.

4 Resultados e discussão

O avanço da expansão da *internet* e a democratização de seu acesso, somados aos esforços da iniciativa privada de grandes empresas de tecnologia, assim como os das próprias comunidades de desenvolvedores de *softwares open source*, possibilitam a viabilidade de novas formas de desenvolver *chatterbots*. Novos produtos e serviços surgem e tornam possíveis as mais variadas arquiteturas de soluções.

Amazon, Microsoft, BRAIN, Google, IBM entre outras, habilitam o uso de suas plataformas de desenvolvimento de *bot* para criação *on-line* ou por meio de SDK disponibilizados nas linguagens de programação mais populares, tais como Node, Ruby, Python ou Java.

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

O SDK permite que os desenvolvedores produzam soluções em suas próprias máquinas, porém a disponibilização do produto final ocorre por meio de *webapis* em servidores remotos da mantenedora da plataforma que disponibiliza o serviço, mas que podem ser consumidas por outros serviços como integração a redes sociais (Facebook, Telegram), por produtos específicos de comunicação tais como Slack ou Rocktchat ou em soluções customizadas como é o contexto do projeto sugerido por este trabalho.

A arquitetura atual deste projeto não contempla nenhum módulo ou ferramenta de gestão de *logs* (registros e outros dados de eventos) e é uma sugestão de melhoria para futuras versões do *bot*.

Os *logs* de negócios são recursos de informações que fornecem possibilidades de futuras melhorias para o *bot*.

Foram desenvolvidos três algoritmos que são capazes de recuperar informações do AVA, do SIGA e do site da Fatec. A coleta de informações é dividida em tarefas assíncronas e são processadas por *workers* independentes que por meio de técnicas de *scraping* e *crawling* populam a base de dados utilizada pelo servidor de *backend*. Porém, se cada uma das fontes de informações disponibilizasse acesso via *webapi*, os dados poderiam ser consumidos em tempo real, diminuindo o tempo de acesso aos dados e reduzindo recursos de infraestrutura e tornaria a arquitetura mais simples.

A abordagem deste trabalho tomou como referência a Fatec Franca, no entanto o projeto pode ser readaptado e implementado para outras instituições de ensino substituindo, removendo ou adicionando os WorkerCrawler tanto quanto forem necessário assim como adicionando novas integrações via WebApis no MCI.

Considerações Finais

As experiências acumuladas durante a execução deste projeto permitiram entender que o desenvolvimento de novas abordagens para a implementação de *chatterbots* estão em contínua evolução. Essas distintas formas de soluções possibilitam que o desenvolvimento de *chatterbot* estejam cada vez mais presentes nos contextos onde os aspectos de comunicação humana tem maior eficiência quando comparados a acessos a sistemas e que, devido às oportunidades de menores custos de tempo de desenvolvimento e também complexidade, promovem

a alavancagem de participação geral da tecnologia como opção de escolha para novas soluções.

No projeto MarinaBot, as entradas de dados feitas pelos usuários são processadas pelo IBM Watson Assistent que, ao reconhecer a intenção expressa no conteúdo enviado, escolhe o fluxo de diálogo correto para responder a ele.

Na etapa de planejamento de diálogo, o levantamento com um conjunto das possíveis perguntas dos usuários sobre cada um dos tópicos de abrangência que o *bot* é projetado para resolver deve ser realizado, fornecendo dessa maneira a base de treinamento vinculando as intenções de usuário, sendo que cada tópico corresponde a uma intenção.

Uma vez treinado o *bot* na plataforma, a integração entre o Watson Assistet com o restante da solução é realizada por *apis* disponibilizadas pela IBM. Dessa forma, o *SaaS*, demonstra que as abstrações das estruturas físicas tais como servidores, redes e demais componentes e também das estruturas lógicas como algoritmos, fluxo de processamento de dados e *softwares*, favorecem para que o foco dos desenvolvedores e demais envolvidos no projeto passe a ser o produto final e os resultados que se deseja obter.

O desenvolvimento de um *chatterbot* conforme as descrições deste artigo foi realizado com sucesso, comprovando as facilidades oferecidas pelas plataformas de serviços que abstraem a necessidade de desenvolver um algoritmo de PLN.

No entanto, planejar a inclusão de qualquer novo componente e como ele se relacionam com a solução de forma global é um desafio a ser gerenciado, pois isso dificulta a detecção de anomalias e falhas do *chatterbot*, esses cenários podem ocorrer pois cada um dos componentes representam uma unidade autônoma de processamento sem relação entre si.

Este trabalho também apresentou a possibilidade de extração de informações de vários sistemas existentes na instituição que atualmente não oferecem um mecanismo formal de conexão com outras ferramentas, centralizando-as em uma base de dados centralizada.

Dessa forma, combinando a tecnologia de robôs de conversação com técnicas de extração de informações aliados a interpolação de dados em mensagens de texto, demonstrou-se a capacidade de fornecer um *software* para facilitar o dia a dia dos alunos utilizando os seus *smartphones* e comunicando em linguagem natural.

Revista EduFatec: educação, tecnologia e gestão
V.1 N.2 julho-dezembro/2018

Agradecimento

Agradeço a professora Flávia Herker, pelo texto traduzido, pela amizade sincera e apoio ao longo desses anos; ao professor Roland pelas provocações e várias discussões sobre como abordar o tema, além de ter feito a primeira pergunta que deixou a Marina confusa; também aos professores Traina e Fausto que tiveram fortes influências em como identificar as fontes de informações mais relevantes para os usuários do *bot*.

Agradeço imensamente também a professora doutora Jaqueline Brigladori Pugliesi por ter emprestado o primeiro livro que li sobre Inteligência Artificial, ele foi crucial para despertar meu interesse de estudo nesta área antes mesmo de conceber as ideias deste trabalho e também pelos conselhos e perseverança.

Aos amigos Thiago Moreira e Thiago Silva pelo incentivo e grande ajuda como usuários da versão beta do projeto; e também as amigas Camila, Thuany e Jaqueline, pela parceira, paciência em ouvir e inúmeras sugestões de melhoria no desenvolvimento deste trabalho e do projeto.

Agradeço também a Miriã Vieira Souza , Cleber Costa da Fonseca e a Raquel de Oliveira Souza Cipriano, pelas horas dedicas, as discussões, revisões de ideias e críticas; sem o apoio de dessas pessoas, esse trabalho nunca teria saído do campo da imaginação.

Agradeço aos amigos do Luizalabs pelo fornecimento de materiais, várias discussões técnicas e críticas na definição da arquitetura do projeto, em especial a Eduardo Cortês, Lucas Devitto, Mahelder Carvalho de Melo. A Hurick Krügner e Antônio Carlos do Nascimento Junior pelas ideias e conceitos sobre UI, UX e PWA e por fim, a Thiago Lelis que me ajudou com os primeiros passos sobre os *Chatterbots* quando apresentou uma palestra sobre o tema e demais esclarecimentos ao longo do desenvolvimento do Marinabot.

Meus agradecimentos especiais são para Maria Luísa Cervi Uzun e Cláudio Eduardo Paiva pelas observações, dicas e ponderações durante a etapa de revisão final deste trabalho.

Referências

CHATBOTS MAGAZINE. 2017. Disponível em: <https://chatbotsmagazine.com/about>. Acesso em: 04 dez. 2018.

CHATBOT REPORT 2018: GLOBAL TRENDS AND ANALYSIS. BRAIN. 2017. Disponível em: <https://chatbotsmagazine.com/chatbot-report-2018-global-trends-and-analysis-4d8bbe4d924b>. Acesso em: 21 out. 2018.

FERRUCCI, D.; BROWN, E.; CHU-CARROLL, J.; FAN, J.; GONDEK, D.; KALYANPUR, A. A.; LALLY, A.; MURDOCK, J. W.; NYBERG, E.; PRAGER, J. et al. Building watson: An overview of the deepqa project. **AI magazine**, v. 31, n. 3, p. 59–79, 2010.

GRAÇA, A, G. **O que podemos aprender com Bia, assistente virtual do Bradesco?**. Disponivel em <https://elife.com.br/index.php/2018/08/16/o-que-podemos-aprender-com-bia-assistente-virtual-do-bradesco/>. Acesso em: 04 dez. 2018.

LAVEN, S, J. **What is a Chatterbot?**. Disponivel em <https://www.simonlaven.com/>. Acesso em: 22 out. 2018.

LEONHARDT, M D; CASTRO, D D; DUTRA, R L S; TAROUCO, L M R. ELEKTRA: Um Chatterbot para Uso em Ambiente Educacional. **Revista Renote Novas Tecnologias na Educação**, Rio Grande do Sul; v. 1, n. 2, p. 1-11, 2003.

MICROSOFT CORPORATION, Redmond, WA (US). Rodion Degtyar, Sammamish, WA (US). **METHOD AND APPARATUS FOR BUILDING METADATA DRIVEN SOFTWARE DEVELOPMENT KIT**. Int CI G06F 7700 (2006.01). US 2006/0143148 A1. 22 jan 2008, 29 jun 2006. United States Patent.

MULLOY, B. **Web API Design Crafting Interfaces that Developers Love**. Disponivel em <https://pages.apigee.com/rs/apigee/images/api-design-ebook-2012-03.pdf>. Acesso em: 22 out. 2018.

NIJIM, S; PAGANO, B. **APIs For Dummies, Apigee Special Edition**. St Hoboken: John Wiley & Sons Inc, 2014.

PONTES, G. **Progressive Web Apps: construa aplicações progressivas com React**. 1 ed. Casa do Código, 2018.

ANDRADE R M. Mobile bot: um chatterboteducacional para dispositivos móveis. **Revista Brasileira de Computação Aplicada**, Passo Fundo, v. 4, n. 2, p. 83-91, out. 2012. ISSN 2176-6649.

RODRIGUES, I. V. G. **Desenvolvimento de um motor de busca e comparação na Web**. 2016. 80f. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) - Universidade Nova de Lisboa, Lisboa, Portugal, 2016.

RUSSEL, S; NORVIG, P. **Inteligência Artificial**. 3 ed. Rio de Janeiro: Elsevier, 2013.

SANTOS, M. Scraping e Memória Digital: Identificando as transformações dos portais jornalísticos a partir da coleta automatizada das suas versões. **6º Simpósio Internacional de Ciberjornalismo**.UFMS, Campo Grande, 2015. Disponível em: http://www.ciberjor.ufms.br/ciberjor6/files/2015/03/Ciberjor15MarcioCSantos1.pdf. Acesso em: 28 out. 2018.

THAISE, D. **Conheça a Profª Elektra, um chatbot desenvolvido pela UFRGS**. Disponível em: <https://medium.com/fred-s-a/conhe%C3%A7a-a-prof%C2%AA-elektra-um-chatbot-desenvolvido-pela-ufrgs-aaf8a4fc412>. Acesso em: 28 out. 2018.

TRÄSEL, M. Jornalismo Guiado por Dados: características definidoras e uma proposta de formulação do conceito**. 15º Encontro Nacional de Pesquisadores em Jornalismo** ECA/USP – São Paulo – Novembro de 2017.Disponível em: http://sbpjor.org.br/congresso/index.php/sbpjor/sbpjor2017/paper/viewFile/794/464 Acesso em: 28 out. 2018.