



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

TOBIAS EVANGELISTA

**DESENVOLVIMENTO DE UM CHATTERBOT PARA AUXILIAR EQUIPES
DE SUPORTE TÉCNICO**

**Assis/SP
2020**



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

TOBIAS EVANGELISTA

**DESENVOLVIMENTO DE UM CHATTERBOT PARA AUXILIAR EQUIPES
DE SUPORTE TÉCNICO**

Projeto de pesquisa apresentado ao Curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Tobias Evangelista

Orientador(a): Prof. Dr. Alex Sandro Romeo de Souza Poletto

**Assis/SP
2020**

FICHA CATALOGRÁFICA

E92d EVANGELISTA, Tobias
Desenvolvimento de um chatterbot para auxiliar equipes de
um suporte técnico / Tobias Evangelista. – Assis, 2020.

44p.

Trabalho de conclusão do curso (Ciência da Computação). –
Fundação Educacional do Município de Assis-FEMA

Orientador: Dr. Alex Sandro Romeo de Souza Poletto

1.Chatterbot 2.Inteligência artificial

CDD006.3

DESENVOLVIMENTO DE UM CHATTERBOT PARA AUXILIAR EQUIPES DE SUPORTE TÉCNICO

TOBIAS EVANGELISTA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Prof. Dr. Alex Sandro Romeo de Souza Poletto

Examinador: _____
Prof. Me. Fábio Eder Cardoso

Assis/SP
2020

RESUMO

Cada vez mais o avanço das tecnologias de comunicação e informação se faz mais presente na sociedade desenvolvendo métodos para melhorar a interação entre o ser humano e a máquina da forma mais autêntica possível. Os *chatterbots* são uma das formas em que isso é feito, afinal são ferramentas que simulam uma conversa com um ser humano através de uma interface interativa. A proposta desse trabalho é desenvolver uma aplicação do tipo *chatterbot* utilizando conceitos de processamento de linguagem natural, para auxiliar equipes de suporte técnico a diminuir solicitações de serviço, tendo em vista que muitas solicitações podem ser resolvidas de maneira fácil por pessoas leigas na área de TI. O *chatterbot* auxiliará na resolução do problema ensinando em passos rápidos e fáceis a solução de problemas simples, desta forma a equipe de suporte técnico poderá focar em problemas que realmente necessitam do seu auxílio.

Palavras-chave: Inteligência Artificial, Processamento de Linguagem Natural, *chatterbot*, Suporte Técnico.

ABSTRACT

The advancement of communication and information technologies is increasingly present in society by developing methods to improve the interaction between human beings and machines in the most authentic way possible. Chatterbots are one of the ways in which this is done, after all they are tools that simulate a conversation with a human being through an interactive interface. The purpose of this work is to develop a chatterbot application using natural language processing concepts, for auxiliary teams of technical support to service requests, considering that many requests can be easily resolved by lay people in the IT area. The chatterbot assists in solving problems, taught in quick steps and tries to solve simple problems and, if not successful, a work order request is automatically created, forming a technical support team specialized in problems that are really used for your assistance.

Keywords: Artificial Intelligence, Natural Language Processing, chatterbot, Technical Support.

LISTA DE ILUSTRAÇÕES

Figura 1: Recursos teóricos-metodológicos para o estudo do PLN	13
Figura 2: Terminal Eliza	16
Figura 3: Mapa mental das áreas de conhecimentos	19
Figura 4: Arquitetura- Visão dos componentes usando como exemplo o facebook messenger. ...	20
Figura 5: Fluxo de Conversa	22
Figura 6: Fluxo de <i>chatbot</i>	23
Figura 7: Diagrama de caso de uso	25
Figura 8: Diagrama de classe	26
Figura 9: Tela de login	27
Figura 10: Tela inicial	27
Figura 11: Tela ordens de serviço	28
Figura 12: Tela cadastro de ordem de serviço	29
Figura 13: Tela produtos	30
Figura 14: Tela cadastro de produtos.....	30
Figura 15: Tela solicitar produtos	31
Figura 16: Tela histórico de retiradas	31
Figura 17: Tela de relatórios	32
Figura 18: Tela de criação do assistente.....	33
Figura 19: Exemplos de intenções criadas na Morgana	34
Figura 20: Exemplos de entidades criadas na Morgana.....	34
Figura 21: Árvore de dialogos	35
Figura 22: Código SDK Facebook.....	38
Figura 23: Código chat plugin	38
Figura 24: Chamada chat plugin	38
Figura 25: Tela inicial com messenger aberto.....	39

SUMÁRIO

1. INTRODUÇÃO	09
1.1 OBJETIVOS	10
1.2 JUSTIFICATIVAS	10
1.3 MOTIVAÇÃO	10
1.4 PERSPECTIVAS DE CONTRIBUIÇÃO	10
1.5 METODOLOGIA DE PESQUISA	11
1.6 ESTRUTURA DO TRABALHO.....	11
2. FUNDAMENTAÇÃO TEÓRICA.....	12
2.1 PROCESSAMENTO DE LINGUAGEM NATURAL	12
2.2 SISTEMA DE RESPOSTAS.....	13
2.3 <i>CHATTERBOT</i>	14
2.4 ENGINE <i>CHATTERBOT</i>	16
3. DESENVOLVIMENTO DO CHATTERBOT	18
3.1 ESCOLHA DO TIPO DE <i>CHATTERBOT</i>	18
3.2 DEFINIR AS ÁREAS DE CONHECIMENTO DO <i>CHATTERBOT</i>	18
3.3 DESENVOLVIMENTO DA ARQUITETURA DO <i>CHATTERBOT</i>	19
3.4 IMPLEMENTAÇÃO DO SERVIDOR WEBHOOK	21
3.5 IMPLEMENTAÇÃO DO SERVIDOR CMS	21
3.6 IMPLEMENTAÇÃO DO SERVIDOR WATSON CONVERSATION	21
3.7 IMPLEMENTAÇÃO DO <i>CHATTERBOT</i>	22
4. IMPLEMENTAÇÃO DA APLICAÇÃO	23
4.1 RECURSOS NECESSÁRIOS PARA O DESENVOLVIMENTO.....	23
4.2 ANÁLISE.....	24
4.2.1 DIAGRAMA DE CASO DE USO.....	24
4.2.2 DIAGRAMA DE CLASSES.....	25
4.3 DESENVOLVIMENTO DA APLICAÇÃO.....	26

4.3.1 INTERFACE E FUNCIONALIDADE DO SISTEMA.....	27
4.4 CRIAÇÃO DO <i>CHATTERBOT</i>	32
4.4.1 INTENTS (INTENÇÕES).....	33
4.4.2 ENTITIES (ENTIDADES).....	34
4.4.3 DIALOG (DIALOGO).....	35
4.4.4 DIGRESSIONS (DIGRESSÕES).....	36
4.5 INTEGRAÇÃO IBM WATSON E FACEBOOK.....	36
4.5.1 CONECTAR APLICATIVO AO WATSON.....	36
4.5.2 CONFIGURAR WEBHOOKS DO FACEBOOK.....	37
4.5.3 CONECTAR O ASSISTENTE.....	37
4.6 INTEGRAÇÃO FACEBOOK MESSENGER COM A APLICAÇÃO.....	37
5. CONCLUSÃO	40
5.1 RESULTADOS.....	40
5.2 CONSIDERAÇÕES FINAIS	40
REFERENCIAS BIBLIOGRAFICAS	42

1. INTRODUÇÃO

Linguagem é o sistema através do qual o homem comunica suas ideias e sentimentos, seja através da fala, da escrita ou de outros signos convencionais, sendo a conversa uma das maneiras fundamentais de comunicação aprendida na infância. Para muitos a capacidade dos computadores de processar a linguagem com a mesma habilidade que nós humanos fazemos anunciara a chegada de máquinas verdadeiramente inteligentes (JURAFSJY, 2009). O incentivo ao avanço tecnológico, assim como a percepção das dificuldades possibilita uma melhor disponibilidade dos procedimentos normalmente adotados.

O Processamento de Linguagem Natural (PLN) é a subárea da Inteligência Artificial que estuda a capacidade e as limitações de uma máquina em entender a linguagem dos seres humanos. O objetivo do processamento de linguagem natural é fornecer aos computadores a capacidade de entender e compor textos. “Entender” um texto significa reconhecer o contexto, fazer análise sintática, semântica, léxica e morfológica, criar resumos, extrair informação, interpretar os sentidos, analisar sentimentos e até aprender conceitos com os textos processados (ROSA, 1997) da mesma forma que ocorre em sistemas de responder perguntas, tradução automático ou de criação de sumários. De acordo com Silva (2008), o processamento de linguagem natural possibilitou a adaptação de tecnologias da fala a aplicações e produtos que possibilitam acessibilidade a pessoas com deficiência visual, pessoas com necessidades especiais, idosos com problemas de se comunicar, os mais conhecidos são os sistemas de conversão texto-fala integrados em e-books e leitores de tela para cegos.

Com a crescente expansão da internet, surgiu os *chatbots*, uma aplicação que utiliza processamento de linguagem natural para simular uma conversa, com diferentes propósitos como “amigo virtual”, assistente comercial ou até simular um psicólogo, para que ocorra essa conversa para cada entrada e saída de informação, as sentenças devem ser atualizadas em seus aspectos morfológicos, sintáticos, semânticos e pragmáticos (RICH: KNIGHT, 1993).

Um dos primeiros sistemas da área, Eliza (WEIZENBAUM, 1966) consegue simular o diálogo do usuário como se ele fosse um paciente de um psicólogo. Neste sistema, as técnicas de PNL são simples, porém sua forma de aplicação ainda é bastante popular em aplicações do tipo *chatbot* (PERNA, DELGADO e FINATTO, 2010).

1.1 OBJETIVOS

O objetivo desse trabalho foi estudar e desenvolver uma aplicação do tipo *chatbot* utilizando conceitos de inteligência artificial com as técnicas de processamento de linguagem natural de forma que ela facilite a comunicação entre o usuário e a máquina

1.2 JUSTIFICATIVAS

Apesar de ser parte do cotidiano, a utilização da tecnologia ainda é uma tarefa difícil para grande parte da sociedade. Uma das alternativas para melhorar e facilitar a comunicação entre o usuário e máquina (aplicação) é o uso do processamento de linguagem natural (PLN). De acordo com SANTOS (2001), o PLN (ou o tratamento das línguas por computador) é uma disciplina que se define como a utilização de conhecimentos sobre a língua e a comunicação humana, sendo ela a comunicação com sistemas computacionais ou a comunicação entre seres humanos. Neste sentido, a proposta desse trabalho foi desenvolver uma aplicação baseada em PLN que facilite e melhore atividades cotidianas, seja ela a adaptação de pessoas com necessidades ou tarefas que necessitem de tecnologia.

1.3 MOTIVAÇÃO

Os desenvolvimentos e os resultados dos estudos em PLN nos últimos anos o colocaram na liderança em uma nova dimensão de aplicações futuras, como o desenvolvimento de interfaces inteligentes que podem interagir com o usuário através do processamento de linguagem natural, ou facilitando a interação homem-computador tornando mais flexível o acesso a informações armazenadas.

1.4 PERSPECTIVAS DE CONTRIBUIÇÃO

Através desse trabalho pretende-se contribuir com futuros trabalhos acadêmicos e auxiliar aplicações cotidianas.

1.5 METODOLOGIA DE PESQUISA

Para conseguir desenvolver uma aplicação com processamento de linguagem natural, foi realizada uma pesquisa, no qual a primeira etapa foi marcada pelo levantamento de bibliografia que discorra sobre o tema. Esta etapa foi dada bastante importância, já que ela servirá como base teórica, resultando em um sólido alicerce de conhecimento para as outras etapas do trabalho.

Após o levantamento bibliográfico, foi apresentada explicações sobre os motivos, causas e problemas relacionados ao tema principal do trabalho. Em seguida, foi apresentado o desenvolvimento prático do tema principal do trabalho seguindo os padrões da engenharia de software.

1.6 ESTRUTURA DO TRABALHO

A estrutura deste trabalho foi composta das seguintes partes:

- **Capítulo 1 – Introdução:** Neste capítulo é contextualizada a área de estudo e apresenta os objetivos, justificativas, motivação, perspectivas de contribuição e metodologia de pesquisa para o desenvolvimento deste trabalho.
- **Capítulo 2 – Fundamentação Teórica:** Neste capítulo, é feita a introdução sobre o que é Processamento de Linguagem Natural e a importância da mesma na evolução da tecnologia e introduzir aos sistemas do tipo *chatbot*, apresentando uma breve história e seus tipos encontrados na web.
- **Capítulo 3 – Desenvolvimento do *chatbot*:** Neste capítulo, é apresentado em detalhes o processo de desenvolvimento de um *chatbot*, construção do *chatbot*, sua arquitetura.
- **Capítulo 4 – Implementação da aplicação:** Neste capítulo, é apresentado em detalhes o processo de desenvolvimento da aplicação, desenvolvimento do *chatbot* Morgana e a integração entre as diversas ferramentas.
- **Capítulo 5 – Resultados:** Neste capítulo, é apresentado os resultados da implementação e as considerações sobre o *chatbot*.
- **Capítulo 6 – Conclusão:** Neste capítulo, apresenta as dificuldades encontradas durante a elaboração deste trabalho, e suas principais contribuições.
- **Referências Bibliográficas**

2. FUNDAMENTAÇÃO TEÓRICA

2.1 PROCESSAMENTO DE LINGUAGEM NATURAL

O estudo em Processamento de Linguagem Natural começou na década de 1940 com tentativas de tradução automática (WEAVER, 1949 apud PERNA, DELGADO e FINATTO, 2010). Esses trabalhos eram realizados com a finalidade de quebra de códigos durante a segunda guerra mundial. Mas somente com Chomsky (1957) que surgiu trabalhos relevantes como o livro *Syntactic Structures* (Chomsky, 1957), a Teoria Sintática da Linguagem (Chomsky, 1965) e dos Algoritmos de Parsing (Aho e Ullman, 1972).

Após esses trabalhos o estudo de PLN aumentou e houve a criação de diversas subáreas que são estudadas até os dias atuais. Os trabalhos desenvolvidos nesse período segundo Jurafsky e Martin (2009) podem ser categorizados como em quatro grupos segundo os paradigmas usados: os métodos estocásticos, os métodos baseados em lógica, os métodos de entendimento de linguagem natural, e os métodos de modelagem de discurso.

Os trabalhos do grupo estocásticos são baseados em abordagens estatísticas e utilizam os formalismos com os modelos ocultos de markov, esses métodos estão na base de diversos trabalhos de síntese de fala (RABINER, HUANG, 1986).

Os trabalhos baseados em lógica começaram com os Q-systems e gramáticas metamórficas (Colmeraeur, 1970 e 1975) os percussores da linguagem prolog (Colmeraeur, 1996) e gramáticas definitivas de clausulas (Pereira e Warren, 1980).

Os trabalhos dos grupos baseados em entendimento da linguagem natural continuaram na vertente de entendimento do discurso (PERNA, DELGADO e FINATTO, 2010). Diversos sistemas de diálogos foram derivados desse grupo como ELIZA (Weizenbaum, 1966) e PARRY (Colby, 1973), também surgiu sistemas de reconhecimento de falas como SHRDLU (Winograd, 1971), LUNAR (Woods, 1977).

Os trabalhos do grupo focado na modelagem do estudo estudaram as questões semânticas. Grandes trabalhos feitos nesse grupo visavam diálogos funcionais.

O Processamento de Linguagem Natural é uma área de Ciências da Computação que estuda o desenvolvimento de programas de computador que analisam, reconhecem e geram textos em linguagem humana (PERNA, DELGADO e FINATTO, 2010). Devido à

grande quantidade de ambiguidades o PLN deixa de ser uma atividade trivial, diferenciando assim do processamento das linguagens de programação que são formalizadas para evitar a ambiguidade das palavras.



FIGURA 1: RECURSOS TEÓRICO-METODOLÓGICOS PARA O ESTUDO DO PLN

Fonte: Silva (2006, p.132).

2.2 SISTEMA DE RESPOSTAS

Um sistema de respostas a perguntas é um programa que analisa uma pergunta e retorna uma ou mais respostas com base na pergunta (NEVES, 2017). Para isso é necessário que o sistema entenda qual a intenção do usuário ao fazer a pergunta, seja para saber um local, uma organização ou até mesmo uma data importante.

Os sistemas de respostas a perguntas podem auxiliar diversas áreas em sistemas críticos e pouco tempo ágil, como sistemas de saúde, comerciais e segurança (FERRUCCI et al.,2010). Segundo FERRUCCI et al. (2010) sistemas de respostas a pergunta é atrativo, porem um dos mais desafiadores para a área de computação e inteligência artificial, pois ela requer síntese de recuperação de informação, processamento de linguagem natural, representação de conhecimento, aprendizado de máquina e interface humano-computador.

Neste projeto foi desenvolvido um *chatterbot* como um sistema de resposta a perguntas para responder perguntas frequentes sobre problemas relacionados a informática, usando como base um sistema de ordem de serviço feito para secretarias municipais de educação na área de tecnologia da informação.

2.3 CHATTERBOT

O termo *chatterbot* vem do inglês em que *chatter* significa conversador e *bot* é uma abreviação para *robot* que significa robô (NEVES, 2017). Os *chatterbots* são agentes inteligentes desenvolvidos para simular uma conversa por troca de mensagens de texto (SGANDERLA, FERRARI, GEYER, 2003), eles utilizam as interações entre pessoas para tentar passar a imagem de que o computador possui uma personalidade. Para que isso ocorra o desenvolvimento do *chatterbot* usa como base o “jogo da imitação” (“Immitation Game”) (TURING, 1950) com intuito de induzir a pessoa a acreditar que está se comunicando com outra pessoa.

Para desenvolver um *chatterbot* é necessário conhecer os diferentes tipos de *chatterbots*, desta forma procurar aquela que mais se encaixa no projeto desejado. Segundo KUNZE (2016) existem dois tipos de *chatterbots*, os de utilidade e os direcionados a conteúdo. Os *chatterbots* de utilidades podem realizar tarefas quando solicitado, enquanto os *chatterbots* de conteúdo seguem um roteiro e fornece algum conteúdo aos seus usuários (NEVES, 2017), existem assistentes que conseguem realizar ambas as funções, seja ela ativar um alarme do celular quando solicitado ou responder perguntas climáticas.

Podemos apresentar ELIZA (Weizenbaum, 1966) como um dos primeiros *chatterbot* desenvolvidos, seu objetivo principal era de simular um psicanalista em uma conversa com seu paciente. Eliza espera que o paciente conte seus problemas e o estimula a contar mais através de perguntas, sempre com uma personalidade amável e agradável. Porém, ela possui limitações de memória, pois um robô não possui memória, ou seja, ela não se lembra

o que foi falado e na tentativa de dar mais naturalidade a conversa, suas respostas podem desencorajar o usuário a continuar a conversa.

Abaixo segue trecho de uma conversa com ELIZA (Weizenbaum, 1966) traduzido por Comarella e Lunardi (2008) e uma imagem de seu terminal adaptado pelo site <https://www.masswerk.at/elizabot/eliza.html> que está no ar até a presente data:

Humana: Os homens são todos iguais.

Eliza: De que forma?

Humana: Eles estão sempre nos incomodando com uma coisa ou outra.

Eliza: Você consegue pensar em um exemplo específico?

Humana: Bem, meu namorado me fez vir aqui.

Eliza: O seu namorado fez você vir aqui.

Humana: Ele diz que eu estou quase sempre deprimida.

Eliza: Sinto muito por saber que você está deprimida.

Humana: É verdade. Eu sou infeliz.

Eliza: Você acha que vir aqui lhe ajudará a não ser feliz?

Humana: Preciso de ajuda, com certeza.

Eliza: O que significaria para você conseguir ajuda?

Humana: Talvez eu conseguisse aprender a me entender com minha mãe.

Eliza: Fale-me mais sobre a sua família.

Humana: Minha mãe cuida de mim.

Eliza: Quem mais na sua família cuida de você?

Humana: Meu pai.

Eliza: Seu pai.

Humana: Você se parece com meu pai em algumas coisas.

Eliza: Qual semelhança você vê?

Humana: Você não é muito agressiva, mas eu acho que você não quer que eu perceba isso

Eliza: O que faz você achar que eu não sou muito agressiva?

Humana: Você não discute comigo

Eliza: Por que você não acha que eu não discuto com você?

Humana: Você tem medo de mim.

Eliza: Te agrada acreditar que eu tenho medo de você?

Humana: Meu pai tem medo de todo mundo.

Eliza: O que vem à sua cabeça quando você pensa no seu pai?

Humana: Tiranos.

Eliza: Isso tem alguma coisa a ver com o fato de o seu namorado tê-la feito vir até aqui?

```

ELIZA Terminal

Welcome to

      EEEEE LL      IIII ZZZZZZ  AAAAA
      EE     LL      II     ZZ   AA  AA
      EEEEE LL      II     ZZ   AAAAAA
      EE     LL      II     ZZ   AA  AA
      EEEEE LLLLLL IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Please tell me what's been bothering you.
YOU: █

```

Figura 2: Terminal Eliza

Fonte: <https://www.masswerk.at/elizabot/eliza.html>

2.4 ENGINE CHATTERBOT

O componente mais importante de um *chatterbot* é sua *engine*, ela é encarregada de transformar a linguagem natural em uma ação entendido pela máquina. Para que a precisão dos *chatterbots* tenha um nível aceitável de precisão as *engines* são desenvolvidas utilizando conceitos de processamento de linguagem natural e aprendizado de máquinas.

Neste projeto será usado o IBM WATSON CONVERSATION como *engine*. O Watson conversation utiliza processamento de linguagem natural e aprendizado de máquina para

extrair intenções e entidades de um diálogo, desta forma elaborar uma resposta mais precisa. A seguir será apresentado os conceitos chaves contidos em uma *engine* que são: Entidades, Intenções, Contexto e Diálogo (NEVES, 2017).

- Entidades: Entidades são informações específicas de domínio, extraídas de uma expressão na qual eles mapeiam frases de linguagem natural para sua forma canônica, a fim de entender a intenção (KAR; HALDAR, 2016 apud NEVES, 2017). Elas são necessárias para tomar ações específicas, para o WATSON elas são usadas para treinamento do *chatbot*, elas são agrupadas em domínios de aplicação.
- Intenções: As intenções representam o que o usuário está buscando fazer ou saber com sua mensagem (KAR; HALDAR, 2016 apud NEVES, 2017).
- Contexto: O contexto é usado quando a entrada do usuário é muito vaga ou possui inúmeros significados baseado no histórico da conversa, eles representam a habilidade dos agentes de manter o estado da conversa, desta forma identificar a intenção do usuário (KAR; HALDAR, 2016 apud NEVES, 2017).
- Diálogos: O diálogo utiliza as intenções, as entidades e o contexto da aplicação para retornar uma resposta baseado na entrada do usuário (KAR; HALDAR, 2016 apud NEVES, 2017).

3. Desenvolvimento do *Chatterbot*

Neste capítulo será mostrado os passos que serão usados no desenvolvimento do *chatterbot*.

3.1 Escolha do Tipo de *Chatterbot*

Na seção 2.3 foram apresentados dois tipos de *chatterbot*. Para este projeto que propõe a utilização de uma base de conteúdo na criação de um *chatterbot* para dar suporte a um sistema de ordem de serviço, foi selecionado desenvolver um *chatterbot* misto que utilizará os dois tipos. Com base nas comparações feitas entre utilizar aprendizado de máquina ou usar uma abordagem dirigida a regras, será usado uma abordagem mista.

3.2 Definir as áreas de conhecimento do *chatterbot*

As áreas de conhecimento do *chatterbot* foram divididas em duas categorias: conhecimentos técnicos e conhecimentos extras.

A área de conhecimento técnico contém as informações diretamente ligadas ao sistema de ordem de serviço e informática. Essa base de conhecimento foi extraída diretamente do sistema autoral desenvolvido.

Com o intuito de tornar o comportamento do *chatterbot* mais parecido com o comportamento humano, a área de conhecimento extra foi criada. Essa área consiste em uma série de subáreas externas ao sistema, que tornam o *chatterbot* mais social. A Figura 3 mostra um mapa mental criado para definir cada área de conhecimento.

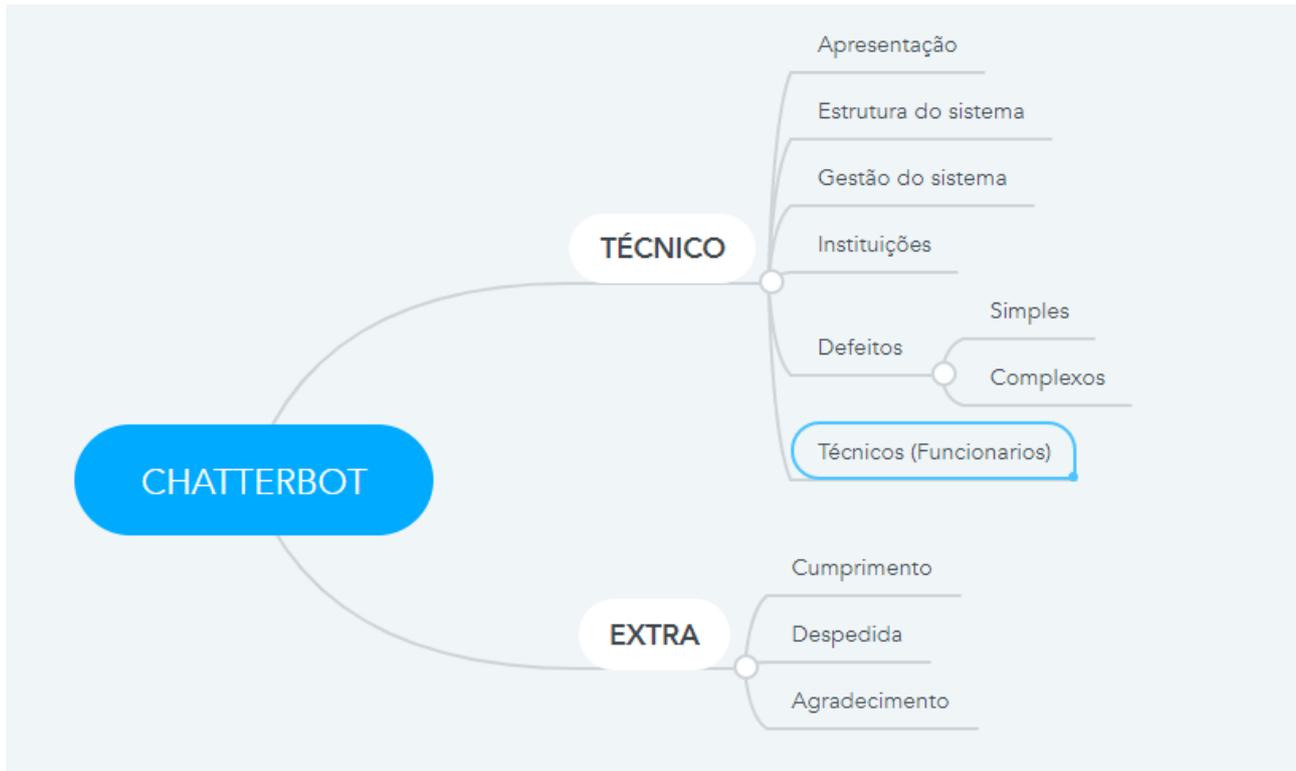


FIGURA 3: MAPA MENTAL DAS ÁREAS DE CONHECIMENTO

Fonte: O Autor

3.3 Desenvolvimento da arquitetura do *chatbot*

Para o desenvolvimento do *chatbot* três módulos são essenciais: uma interface humano-computador onde será feita a comunicação do usuário com o *chatbot*, um sistema de gerenciamento de conteúdo (CMS) para gerenciar informações variáveis e o próprio *chatbot* (NEVES,2017).

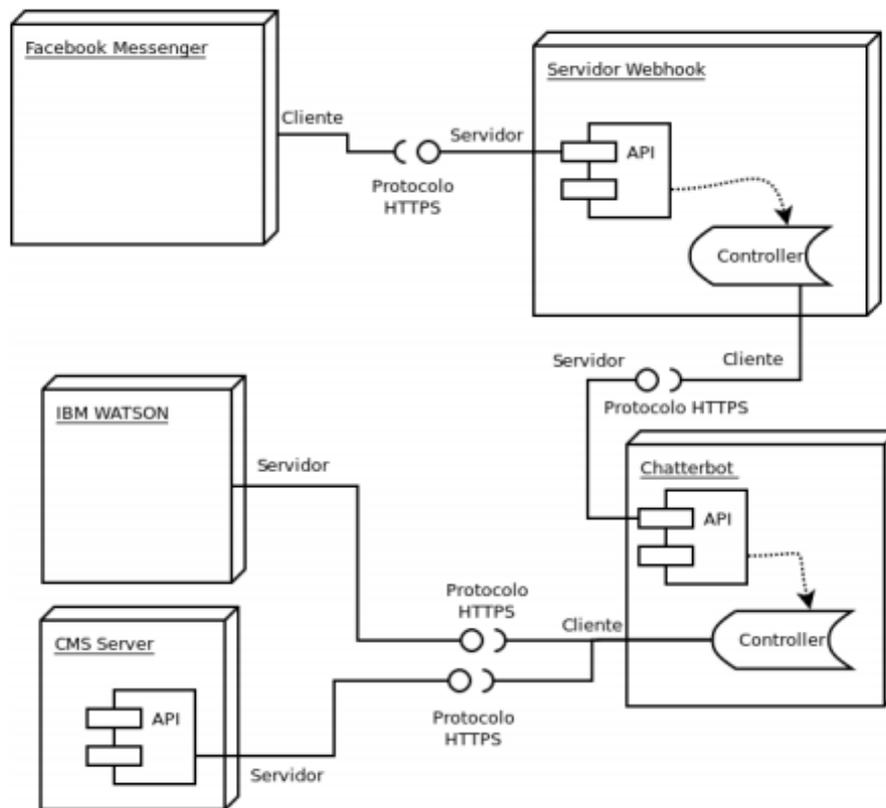


FIGURA 4: ARQUITETURA- VISÃO DOS COMPONENTES USANDO COMO EXEMPLO O FACEBOOK MESSENGER

Fonte: NEVES, 2017

Webhook é a forma de receber informações entre dois sistemas de uma forma passiva, ou seja, quando um evento acontece. Foi utilizado para fazer o papel de interface humano-computador através do Facebook Messenger. Para desenvolver uma aplicação para o Facebook Messenger é necessário que se implemente um servidor *webhook*, que implemente a interface de API necessária para que o Messenger seja integrado com a aplicação (NEVES, 2017).

O CMS (sistema) foi desenvolvido utilizando o framework Django com uma API REST para a comunicação com o *chatterbot* e uma interface web para a inserção dos dados pelos usuários.

Para o *chatterbot* foi criado uma aplicação em Python que implementa uma estrutura básica do *chatterbot* e se comunica com o IBM Watson Conversation e com o CMS para elaborar as respostas do *chatterbot*.

3.4 IMPLEMENTAÇÃO DO SERVIDOR WEBHOOK

O *Webhook* é o intermediário entre o Facebook Messenger e o *chatbot*. Seu funcionamento é simples, ele fica aguardando requisições do tipo POST vinda do Facebook Messenger, e para cada mensagem recebida, ele envia a mensagem ao *chatbot* e devolve a resposta recebida para o mesmo. O *Webhook* no projeto consistirá em uma aplicação em Python, rodando em um servidor Ubuntu.

3.5 IMPLEMENTAÇÃO DO SERVIDOR CMS

O servidor CMS foi desenvolvido para que informações sobre a Gestão e Técnicos que podem mudar com frequência possam ser gerenciadas de maneira fácil, desta forma pessoas leigas em computação. Informações como nome do gestor, nome dos técnicos, telefone do setor, dentre outras foram as informações gerenciadas pelo CMS. O servidor que será usado será o Ubuntu e o CMS desenvolvido em Python utilizando o framework Django 2.2.

3.6 IMPLEMENTAÇÃO DO WATSON CONVERSATION

Esse passo foi um dos mais importantes, pois é nele que as regras da *engine* serão definidas. A IBM oferece uma interface gráfica para a definição das regras, desta forma facilita o desenvolvimento.

Primeiro foram definidas as possíveis intenções dos usuários, baseado no mapa mental apresentado na Figura 3. Entender qual é a intenção do usuário já é um passo bem importante no desenvolvimento do diálogo, porém além de conseguir saber a intenção, é importante saber sobre qual entidade aquela intenção está relacionada (NEVES, 2017). Então serão definidas as entidades do *chatbot*.

Para que haja a união entre a intenção do usuário com a entidade relacionada terá que ser criado o fluxo de diálogos. O fluxo de diálogos funcionará como uma árvore de decisões, onde as condições serão avaliadas como verdadeiras ou falsas para que desta forma seja determinado o fluxo do diálogo.

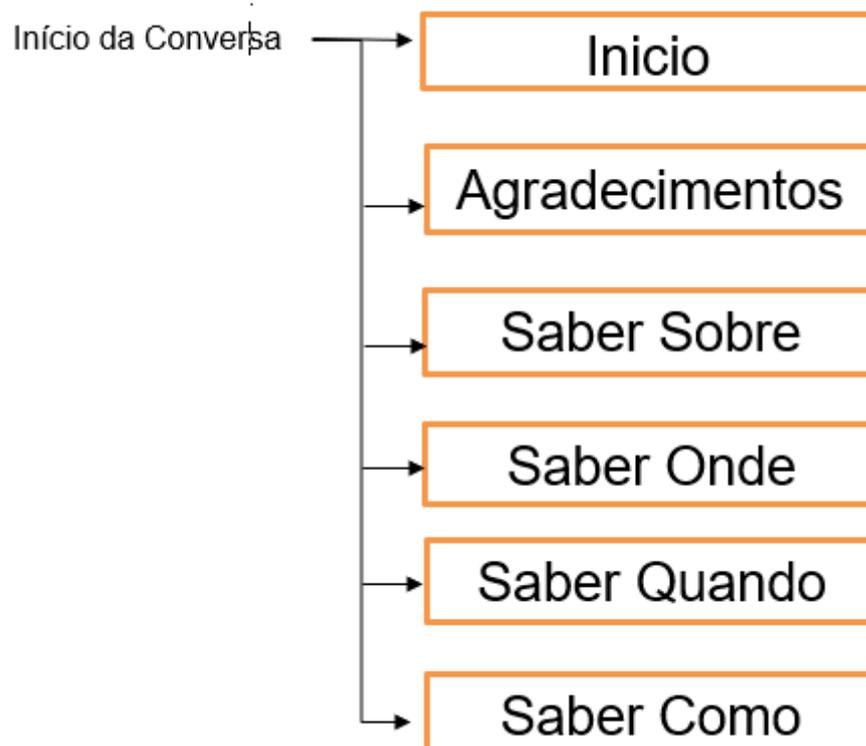


FIGURA 5: FLUXO DE CONVERSA

Fonte: adaptado de NEVES (2017)

Para cada nó no fluxo de diálogo, terá que existir um campo chamado *trigger*, que é um gatilho condicional que definirá se aquele diálogo entra naquele nó ou não. Por exemplo, se for definido o trigger que representa a intenção “Saber Sobre” no nó saber sobre. Então, durante o diálogo, quando for identificada a intenção do usuário de saber sobre algo, o fluxo do diálogo será desviado para o nó “Saber Sobre” (NEVES, 2017).

3.7 IMPLEMENTAÇÃO DO CHATTERBOT

O servidor do *chatterbot* foi desenvolvido na plataforma IBM Cloud e o front end foi feito utilizando o Messenger através da plataforma facebook for developers. O *chatterbot* ficará em constante aguardo por mensagens. Ao receber uma mensagem ele encaminhará a mensagem para o IBM Watson Conversation (NEVES, 2017). Baseado nas regras definidas do Watson, ele elabora uma resposta e retornar a resposta do *chatterbot*. Uma vez que o *chatterbot* receber a resposta será feita uma análise se uma consulta ao CMS é necessária.

Caso seja, uma consulta é feita ao CMS para obter as informações necessárias e em seguida, uma resposta é retornada.



FIGURA 6: FLUXO DE *CHATTERBOT*

Fonte: NEVES, 2017

4. Implementação da Aplicação

4.1 Recursos necessários para o desenvolvimento

Para o desenvolvimento foi utilizado os seguintes recursos:

- Notebook Samsung Core i5, 8GB de memória, HD 1TB.
- Visual Studio Code

- MYSQL Enterprise Edition
- IBM Cloud

4.2 Análise

Como ambiente para o estudo de casos, este trabalho desenvolveu um sistema de gerenciamento de uma equipe de TI, onde é possível gerenciar ordens de serviço e solicitação de produtos, substituindo o atendimento presencial e/ou por telefone, desta forma ocupando o tempo de parte da equipe, entre os principais problemas estão:

- Desorganização
- Andamento
- Histórico
- Desperdício de mão de obra técnica

Para fazer a análise da aplicação foi utilizado as ferramentas de modelagem da UML (Unified Modeling Language).

4.2.1 Diagrama de caso de uso

O diagrama de caso de uso representa como os casos de uso interagem entre si no sistema e com os usuários (atores).

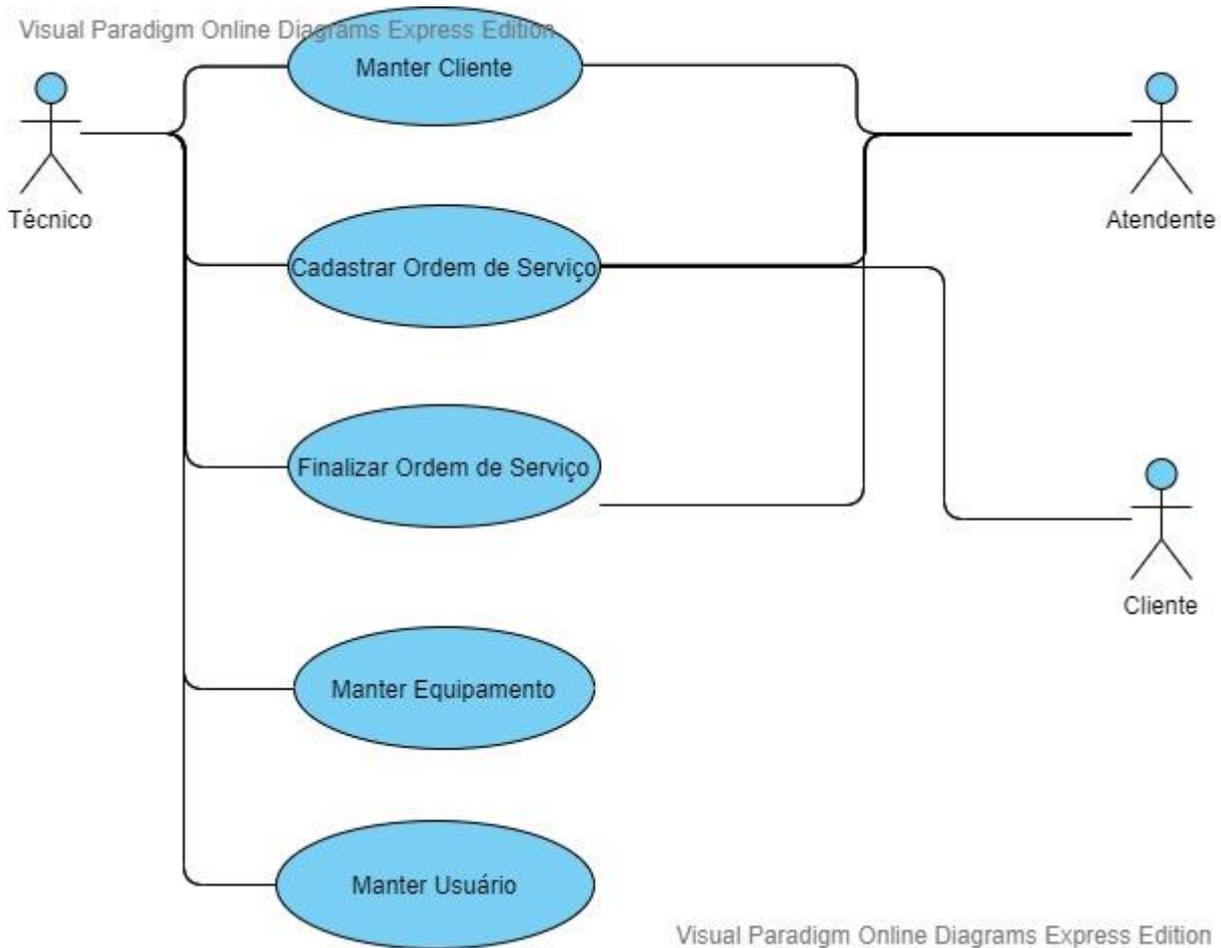


FIGURA 7: DIAGRAMA DE CASO DE USO

Fonte: O autor

4.2.2 Diagrama de classe

O diagrama de classe ilustra graficamente como será a estrutura do software e ilustrar as classes do software instanciadas.

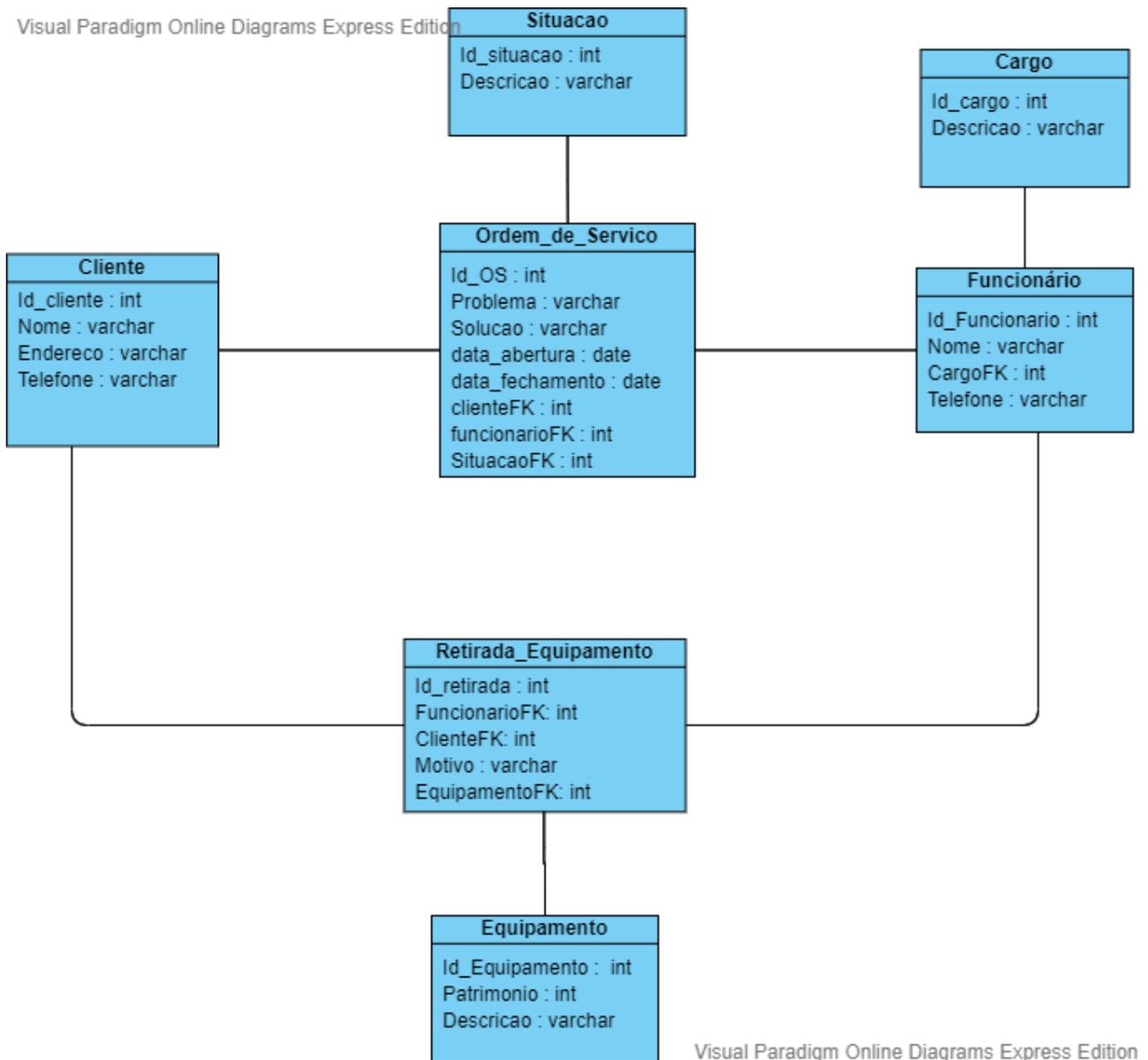


FIGURA 8: DIAGRAMA DE CLASSE

Fonte: O autor

4.3 Desenvolvimento da Aplicação

Para o desenvolvimento da aplicação foi utilizado a linguagem de programação Python 3.7 e o framework Django 2.2.

O Django utiliza o padrão de projeto MTV (model, template e *view*), onde na camada model acontece a interação com o banco de dados, a camada template é feita a interação do usuário com a aplicação e a camada *view* é a responsável pelas regras de negócio.

4.3.1 Interface e Funcionalidade do Sistema

Ao entrar no sistema, o usuário deverá fazer o login, inserindo seu usuário e sua senha, caso tenha esquecido a senha ou não tenha cadastro deverá apertar no botão “Esqueci minha senha” ou “Registrar”. Após o usuário ter efetuado o login com sucesso, ele é redirecionado para página principal do sistema como mostra na figura 10, onde ele terá várias funcionalidades do sistema.

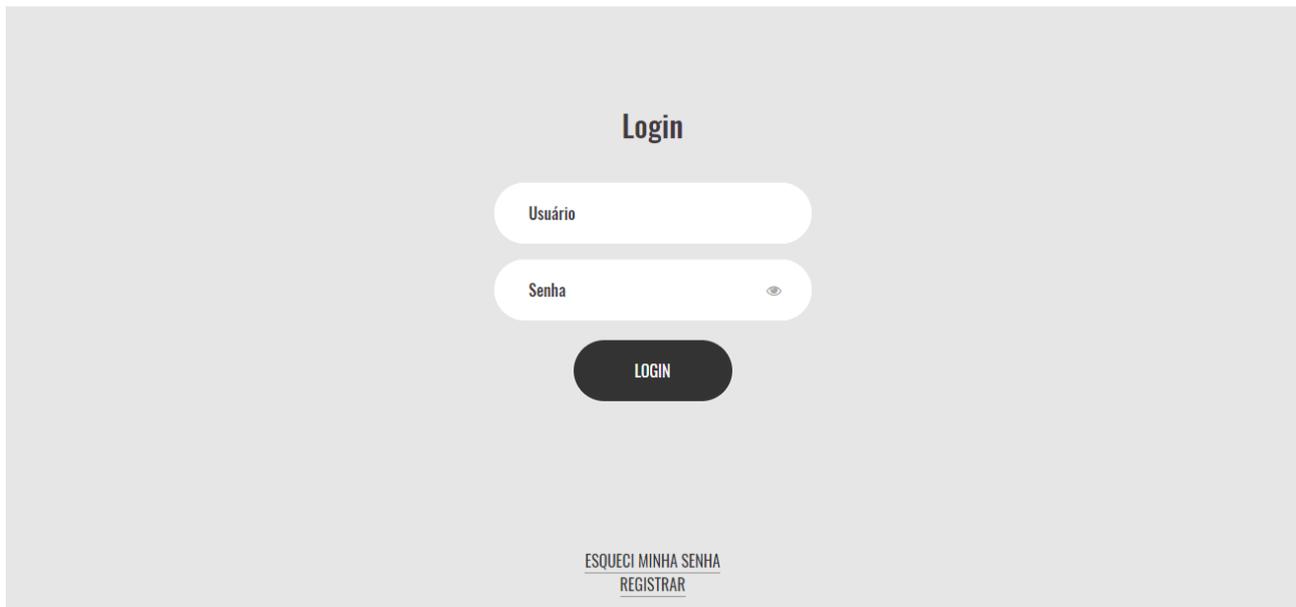


FIGURA 9: TELA DE LOGIN

Fonte: O autor

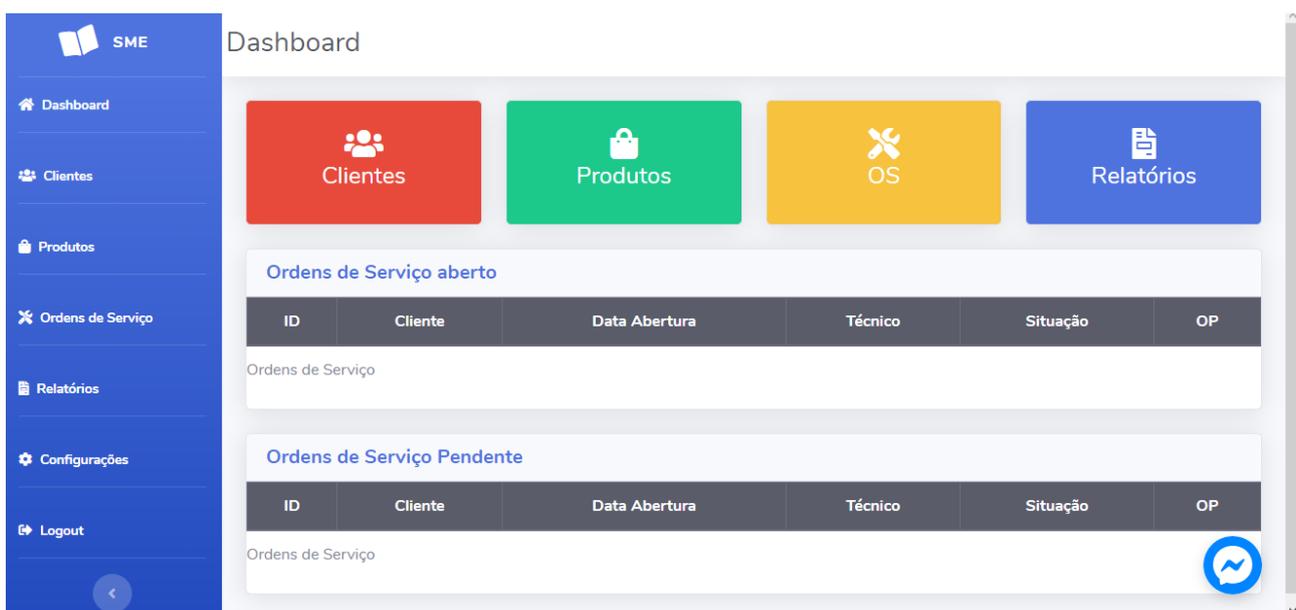


FIGURA 10: TELA INICIAL

Fonte: O autor

Na página inicial do sistema, é mostrado as funcionalidades do sistema e as ordens de serviço em aberto e pendentes, abaixo há um ícone do Messenger, que quando apertado o usuário irá começar a se comunicar com o assistente. O assistente irá auxiliar o usuário em dúvidas sobre as funcionalidades do sistema e realizar algumas operações de maneira simples e direta, simulando uma conversa humana.

Se o usuário selecionar “Ordem de Serviço” na página inicial ele será redirecionado para a página de ordem de serviço que contem tabelas com ordens de serviços abertas, pendentes e fechadas, conforme a Figura 11. Se o usuário quiser procurar uma ordem de serviço de um cliente específico ele deve colocar o nome do cliente no campo “Buscar” e o sistema irá filtrar apenas o cliente informado, caso ele queira criar uma nova ordem de serviço ele irá selecionar o botão “Adicionar” e será redirecionado para a página de cadastro de ordem de serviço, conforme a Figura 12.

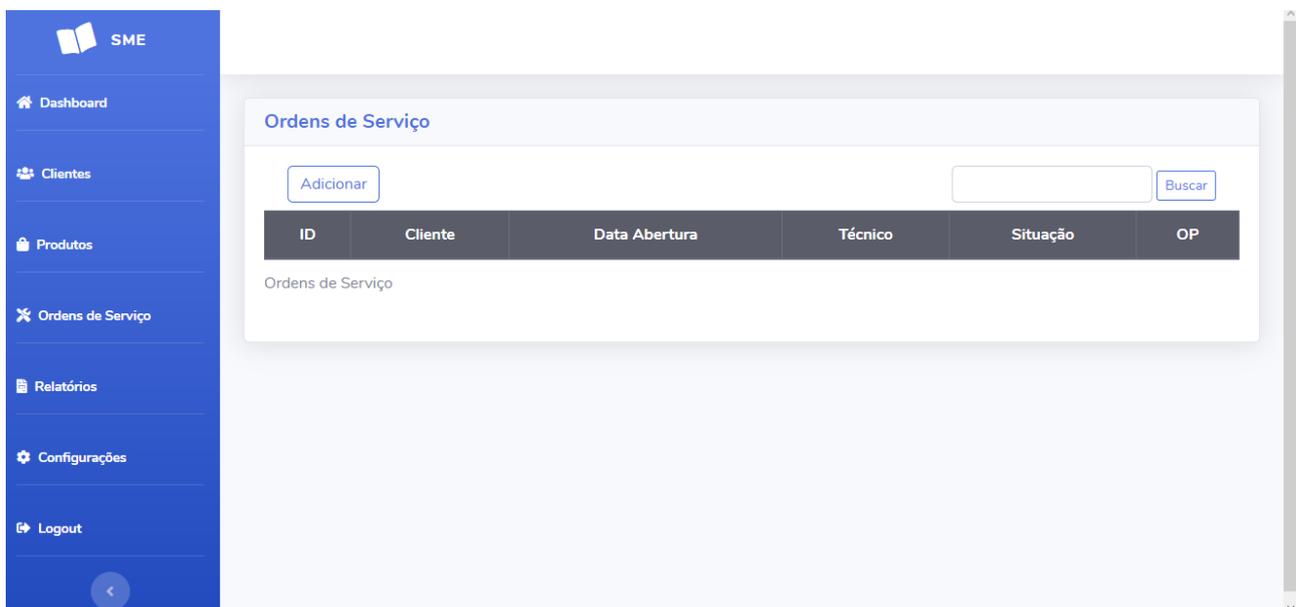


FIGURA 11: TELA DE ORDENS DE SERVIÇO

Fonte: O autor

The image shows a web application interface for registering a service order. On the left is a blue sidebar menu with the following items: Dashboard, Clientes, Produtos, Ordens de Serviço, Relatórios, Configurações, and Logout. The main content area is titled 'Cadastro Ordem de Serviço' and contains the following form fields: 'Problema' (text input), 'Solucao' (text input), 'Data abertura' (date input with a calendar icon), 'Data fechamento' (date input with a calendar icon), 'Cliente' (dropdown menu), 'Tecnico' (dropdown menu), and 'Situacao' (dropdown menu). A 'Salvar' button is located at the bottom of the form.

FIGURA 12: TELA DE CADASTRO DE ORDEM DE SERVIÇO

Fonte: O autor

Na página inicial caso o usuário selecione a opção “Produtos” ele irá ser redirecionado para a página de produtos, onde irá ter uma tabela com os produtos cadastrados e sua quantidade restante no estoque, como mostra a Figura 13. Se o usuário quiser adicionar um novo produto, deverá selecionar a opção “Adicionar”, que irá redirecionar para página de cadastro de produtos, conforme a Figura 14. Se o usuário quiser fazer solicitação de produto, deverá selecionar a opção “Solicitar” na página de produtos, que irá redirecionar para a página de solicitação de produto, como mostra a Figura 15. Ainda na página de produtos, o usuário também terá a opção de ver o histórico de produtos retirados/solicitados, selecionando a opção “Histórico”, que irá redirecionar para a página de histórico de produtos, como mostra a Figura 16.

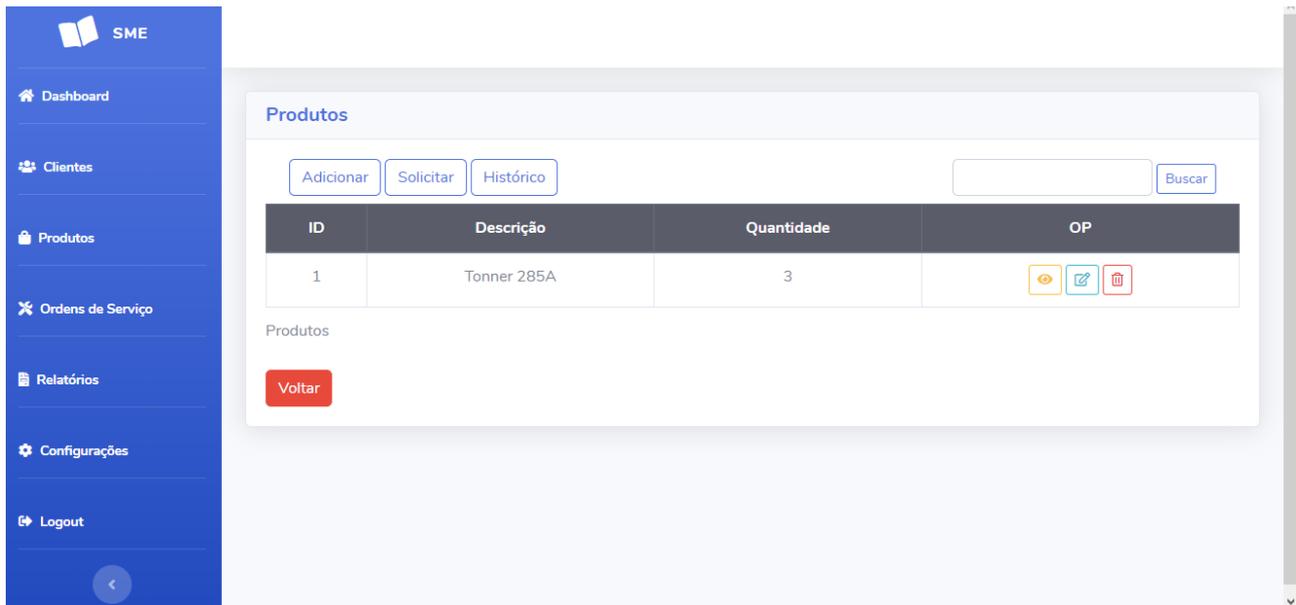


FIGURA 13: TELA DE PRODUTOS

Fonte: O autor

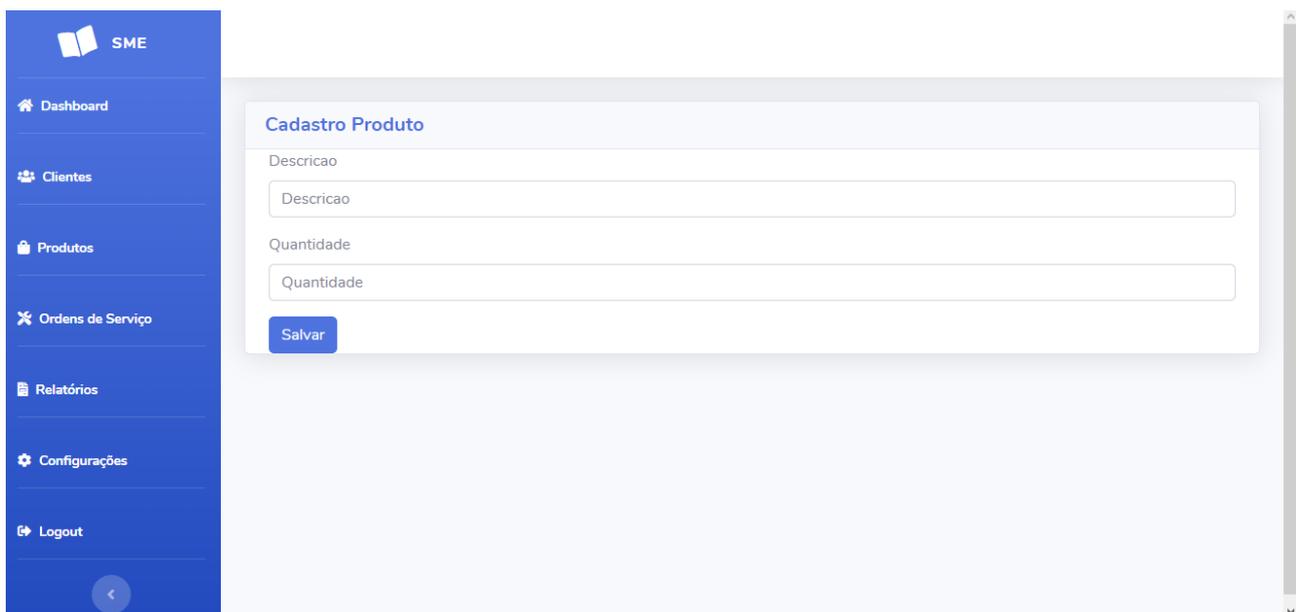


FIGURA 14: TELA DE CADASTRO DE PRODUTOS

Fonte: O autor

The screenshot shows a mobile application interface for 'SME'. On the left is a blue sidebar menu with options: Dashboard, Clientes, Produtos, Ordens de Serviço, Relatórios, Configurações, and Logout. The main content area is titled 'Retirada de Produto' and contains a form with the following fields:

- Produto: A dropdown menu with a dashed line indicating a selection.
- Quantidade: A text input field with the placeholder 'Quantidade'.
- Motivo: A text input field with the placeholder 'Motivo'.
- Cliente: A dropdown menu with a dashed line.
- Funcionario: A dropdown menu with a dashed line.

 At the bottom of the form are two buttons: 'Salvar' (blue) and 'Voltar' (red).

FIGURA 15: TELA SOLICITAR PRODUTO

Fonte: O autor

The screenshot shows the 'Retiradas' (Withdrawals) history page. It features a table with the following structure:

ID	Produto	Cliente	Funcionario	Quantidade
Retiradas				

 Below the table is a red 'Voltar' button. The sidebar menu on the left is identical to the previous screenshot.

FIGURA 16: HISTÓRICO DE RETIRADA

Fonte: O autor

Na página inicial o usuário poderá selecionar a opção relatórios, que irá redirecionar para a tela de relatórios, onde poderá ver um relatório de ordens de serviços e produtos,

conforme a figura 17, o relatório será feito no formato PDF para que o usuário possa fazer o download ou impressão do mesmo.

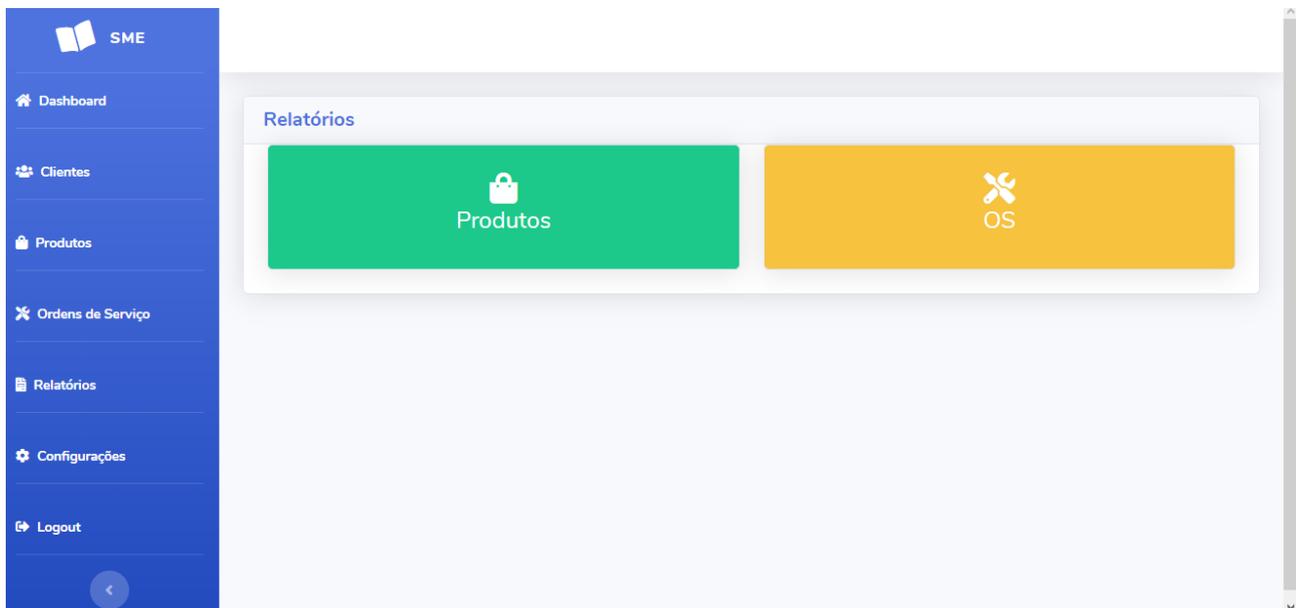


FIGURA 17: TELA DE RELATÓRIOS

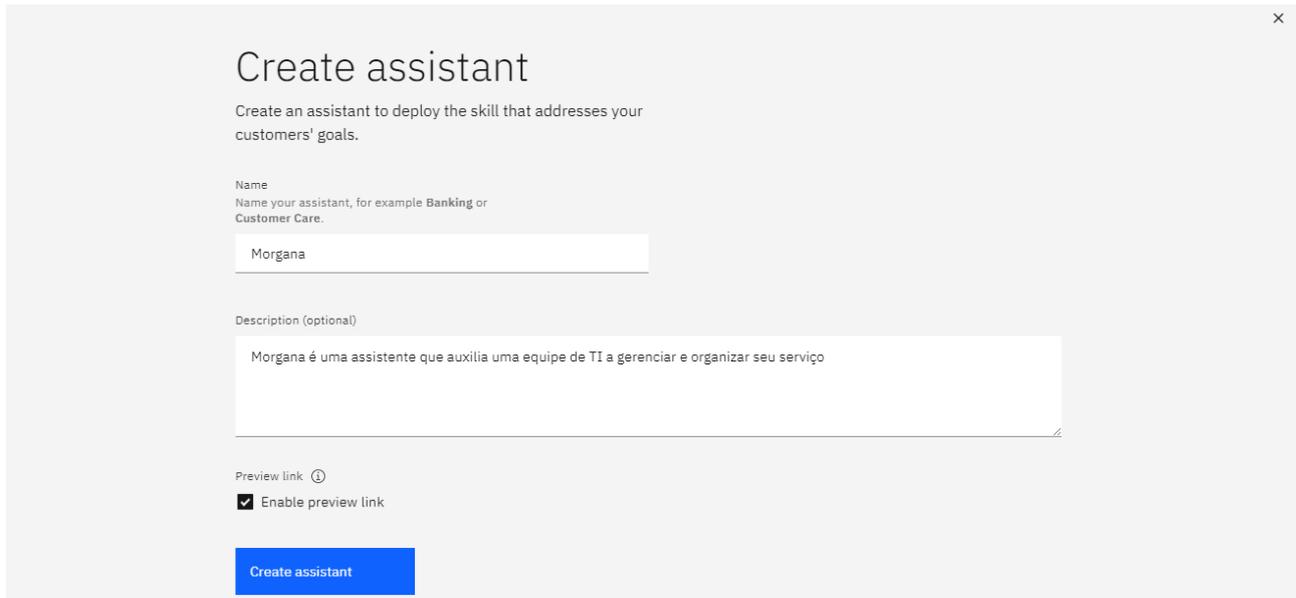
Fonte: O autor

4.4 Criação do *Chatterbot*

Para esse trabalho foi escolhido desenvolver o *chatterbot* utilizando as ferramentas disponibilizadas na IBM Cloud, devido a sua interface gráfica para definição de regras e trabalhar com computação cognitiva, facilitando a criação de um assistente.

Para criar um assistente através da IBM Watson é necessário criar uma conta na plataforma IBM Cloud. A IBM Cloud é o serviço da nuvem da IBM que contém diversos componentes que trabalham juntos para fornecer uma experiência de nuvem consistente e confiável.

Após a criação da conta IBM Cloud, o primeiro passo para a criação de um assistente foi definir seu nome e opcionalmente fazer uma descrição do mesmo, como apresentado na Figura 18. Para esse projeto foi escolhido o nome de “Morgana” e foi feito uma breve descrição: “Morgana é uma assistente que auxilia uma equipe de TI a gerenciar e organizar seus serviços”.



Create assistant

Create an assistant to deploy the skill that addresses your customers' goals.

Name
Name your assistant, for example Banking or Customer Care.

Morgana

Description (optional)

Morgana é uma assistente que auxilia uma equipe de TI a gerenciar e organizar seu serviço

Preview link ⓘ

Enable preview link

Create assistant

FIGURA 18: TELA DE CRIAÇÃO DO ASSISTENTE

Fonte: O autor

Depois de criado o assistente, foi necessário treina-lo com quatro funcionalidades para o que ele cumpra seu propósito. São elas: *Intents* (intenções), *Entities* (entidades), Dialog (diálogo), Digressions (Digressões).

4.4.1 *Intents* (intenções)

Intenções ou *Intents* são ações ligadas as perguntas feitas pelo usuário, ou seja, é o que o usuário procura ao se comunicar com o sistema (SILVA,2018), assim como na linguagem natural, a forma de comunicação para um mesmo propósito é diversa e é impossível cadastrar todas as formas, para resolver isso, o assistente utiliza intenções previamente cadastradas nas quais são fornecidos exemplos de frases, utilizando PLN o sistema conseguirá identificar outros exemplos e relacionar com intenções já existentes.

Durante o desenvolvimento desse trabalho foram criadas intenções que um cliente pode ter ao manipular o sistema o sistema de solicitação de produtos, criação de ordens de serviço e/ou problemas relacionados a área de TI.

<input type="checkbox"/>	Intents (5) ↑	Description	Modified ↑↓	Conflicts ↑↓	Examples ↑↓
<input type="checkbox"/>	#clienteos	Problema com cliente	a few seconds ago		1
<input type="checkbox"/>	#dataos	Problema com data	a few seconds ago		1
<input type="checkbox"/>	#OrdemService	intenções relacionados a orde...	a few seconds ago		1
<input type="checkbox"/>	#produtos	Intenções relacionados a prod...	a minute ago		1
<input type="checkbox"/>	#Saudação	Saudações feitas pelos usuários	an hour ago		4

FIGURA 19: EXEMPLOS DE INTENÇÕES CRIADA NA MORGANA

Fonte: O autor

4.4.2 *Entities* (entidades)

Entidades ou *Entities* são complementos de informação, o assistente irá identificar esses itens através da interação do usuário e extrai-los, desta forma fazer uma abordagem diferente para a pergunta (SILVA, 2018). Por exemplo, se o usuário iniciar a interação com a seguinte sentença: “Preciso de tonner para impressora HP”, através da entidade ele reconhece que “tonner” é um produto do estoque, desta forma refinar o seu conhecimento, ou seja, armazenar essa informação para usar futuramente ela, melhorando a interação com o usuário. Entidades com valores parecidos devem ser agrupadas e os sinônimos devem ser relacionados a palavra, assim é possível reconhecer entidades com valores iguais.

Durante o desenvolvimento foi criado várias entidades para armazenamento de funções que auxiliam cadastros e solicitações.

<input type="checkbox"/>	Entity (4) ↑	Values	Modified ↑↓
<input type="checkbox"/>	@ordem	ordem de serviço	an hour ago
<input type="checkbox"/>	@ordemservice	Editar Ordem de serviço, Salvar Ordem de Serviço, Encontrar Cliente, Encontra...	19 hours ago
<input type="checkbox"/>	@problema	ordem de serviço, produto	a day ago
<input type="checkbox"/>	@produto	produto	44 minutes ago

FIGURA 20: EXEMPLOS DE ENTIDADES CRIADAS NA MORGANA

Fonte: O autor

4.4.3 Dialog (diálogo)

Dialogo ou dialog utilizam as intenções de entrada do usuário, juntamente com o contexto para dar uma resposta útil, a resposta pode ser para uma pergunta ou solicitação, dependendo do contexto. Os diálogos funcionam através de aplicação de regras simples como tratamentos e condicionais de decisão (SILVA, 2018), quando o usuário realizar uma entrada o assistente realiza uma busca na estrutura de dialogo pré-definida procurando referência que pode ser aplicada, dessa forma fornecer uma resposta, se o assistente não encontrar nenhuma correspondência na arvore de decisão, a linha execução vai para a entrada padrão não validada. Os diálogos variam dependendo do propósito de cada assistente, para o desenvolvimento desse trabalho foi criado uma arvore de dialogo especifico, conforme mostra a Figura 21.

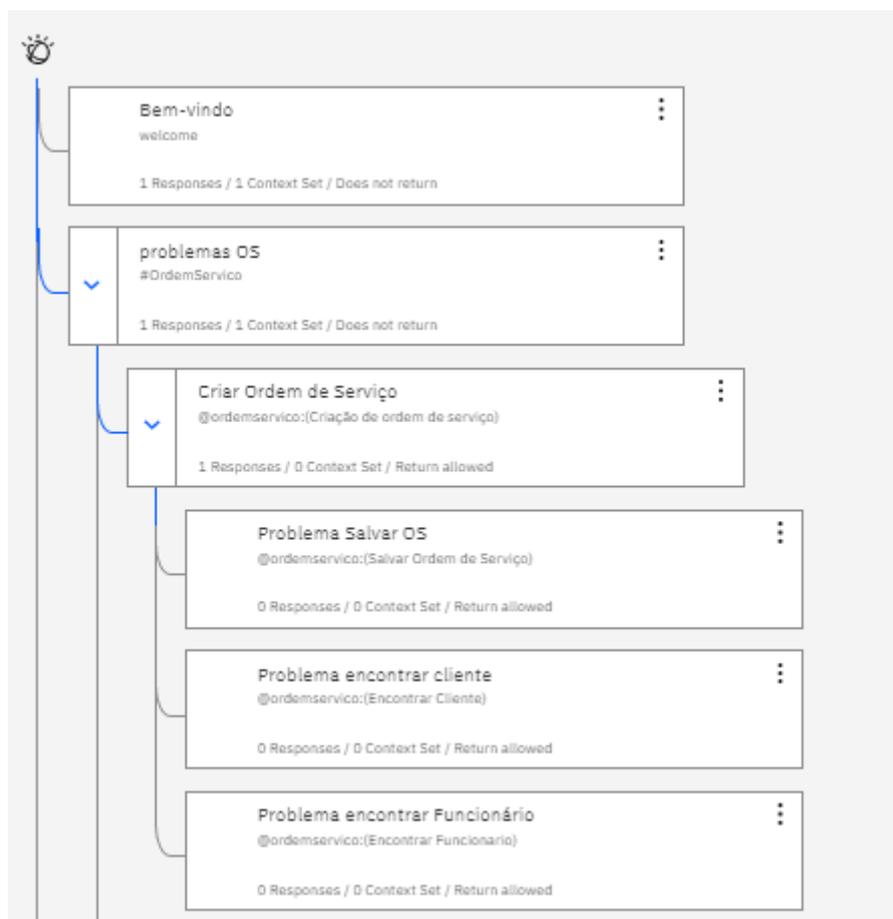


FIGURA 21: ARVORE DE DIALOGO

Fonte: O autor

4.4.4 Digressions (Digressões)

Digressões ocorrem quando o usuário está em um fluxo de nó de diálogo e muda abruptamente para outro conteúdo, ou seja, para outra subarvore ou outra raiz (SILVA, 2018). Por exemplo, caso o usuário esteja no meio de uma abertura de ordem de serviço e então o *bot* pergunta qual o nome do cliente cadastrado e o usuário interrompe o fluxo com uma pergunta sobre a solicitação de produtos.

Através das digressões o Watson cria pontos de volta, assim conseguindo dar andamento no novo fluxo e ao termina-lo, continuar o antigo fluxo de onde foi parado. Na prática os pontos de digressões servem para deixar a conversa com sensação de uma conversa real.

4.5 Integração IBM Watson e Facebook

Com o *chatbot* já criado, é necessário fazer a integração do mesmo com alguma plataforma de mensagem, a IBM disponibiliza a ferramenta *integrations* (integração) para facilitar a integração com algumas plataformas como Facebook e Telegram. Para esse trabalho foi escolhido o Facebook Messenger devido a sua facilidade de integração ao sistema.

Para fazer a integração entre do *chatbot* e Facebook é necessário cumprir pré-requisitos:

- Ter uma conta na IBM Cloud.
- Ter uma conta no Facebook.
- Ter uma conta no Facebook for Developers.

A IBM disponibiliza o passo a passo da integração através da ferramenta *integrations*, os passos para a integração levando em consideração o *bot* já criado são:

- Criar uma página no Facebook.
- Adicionar um novo aplicativo no Facebook for Developers.
- Conectar o aplicativo ao Watson.
- Configurar *webhooks* do Facebook.
- Conectar o assistente.

4.5.1 Conectar o aplicativo ao Watson

Depois de criado a página e o aplicativo no Facebook for Developers, é necessário fazer a conexão da página criada ao Watson, o Facebook for Developers gera um token de autenticação da página para conectar a mesma ao Watson. Após copiar o token coloque-o na área de autenticação do Watson.

4.5.2 Configurar webhooks do Facebook

O *webhook* é o núcleo da experiência com o Messenger, é o responsável por receber, processar e enviar as mensagens, os *webhooks* são enviados por meio de HTTPS. As mensagens recebidas disparam eventos que são atualizados através da integração de Webhooks. Após receber e validar as mensagens, as solicitações de validação e notificações de webhook são enviadas para *url* de retorno de chamada e o Facebook enviará também um token, como parte da verificação da *url* de retorno de chamada. A *url* e o token são gerados automaticamente pelo Watson, onde é necessário apenas copiar e colocar no lugar solicitado.

4.5.3 Conectar o assistente

Após configurado os *webhooks*, na página do Facebook for Developers, você irá selecionar a página criada anteriormente, desta forma será feito a integração do assistente a página do Facebook, até o momento não é suportado múltiplas-páginas, por isso a importância de selecionar a página correta em que será feito a integração.

4.6 Integração Facebook Messenger com a aplicação

Para a integrar o Messenger a aplicação, foi usado o chat plugin, disponibilizado pelo Facebook for Developers. O chat plugin carrega automaticamente o histórico de bate-papo entre a pessoa e o *bot*, dessa forma, caso o usuário queira, é possível continuar a conversa mesmo após ter saído da aplicação e oferece suporte aos navegadores mais populares.

Para incluir o chat plugin é necessário incluir uma versão do SDK (Software Development Kit) do Facebook javascript que contenha o SDK do chat plugin na página da aplicação em que ele será renderizado. O SDK do Facebook para javascript não precisa que seja feito download ou instalação, é necessário apenas que seja incluído um pequeno pedaço de javascript regular na página html, conforme mostra a Figura 22, o código apresentado

carregará de forma assíncrona o SDK na página, ou seja, ele não irá bloquear o carregamento de outros elementos da página.

```
<script>
  window.fbAsyncInit = function() {
    FB.init({
      appId      : '655217185103445',
      autoLogAppEvents : true,
      xfbml      : true,
      version    : 'v8.0'
    });
  };
</script>
<script async defer crossorigin="anonymous" src="https://connect.facebook.net/pt_BR/sdk.js"></script>
```

FIGURA 22: CÓDIGO SDK FACEBOOK

Fonte: O autor

Depois de incluído o JDK do Facebook javascript, será feito a inclusão do chat plugin a página html através de uma função javascript.

```
<script>
  (function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src = "https://connect.facebook.net/pt_BR/sdk/xfbml.customerchat.js";
    fjs.parentNode.insertBefore(js, fjs);
  })(document, 'script', 'facebook-jssdk');
</script>
```

FIGURA 23: CÓDIGO CHAT PLUGIN

Fonte: O autor

Após incluído o javascript chat plugin na página html, é possível fazer a chamada dele através da classe “class = “fb-customerchat”, como mostra a Figura 24.

```
<div class="fb-customerchat"
  page_id="PAGE_ID"> <!-- ID da pagina do facebook -->
</div>
```

FIGURA 24: CHAMADA CHAT PLUGIN

Fonte: O autor

O chat plugin ficará em todas as telas como um botão com o ícone do Facebook Messenger, para quando pressionado abra a tela de bate-papo, conforme a Figura 25.

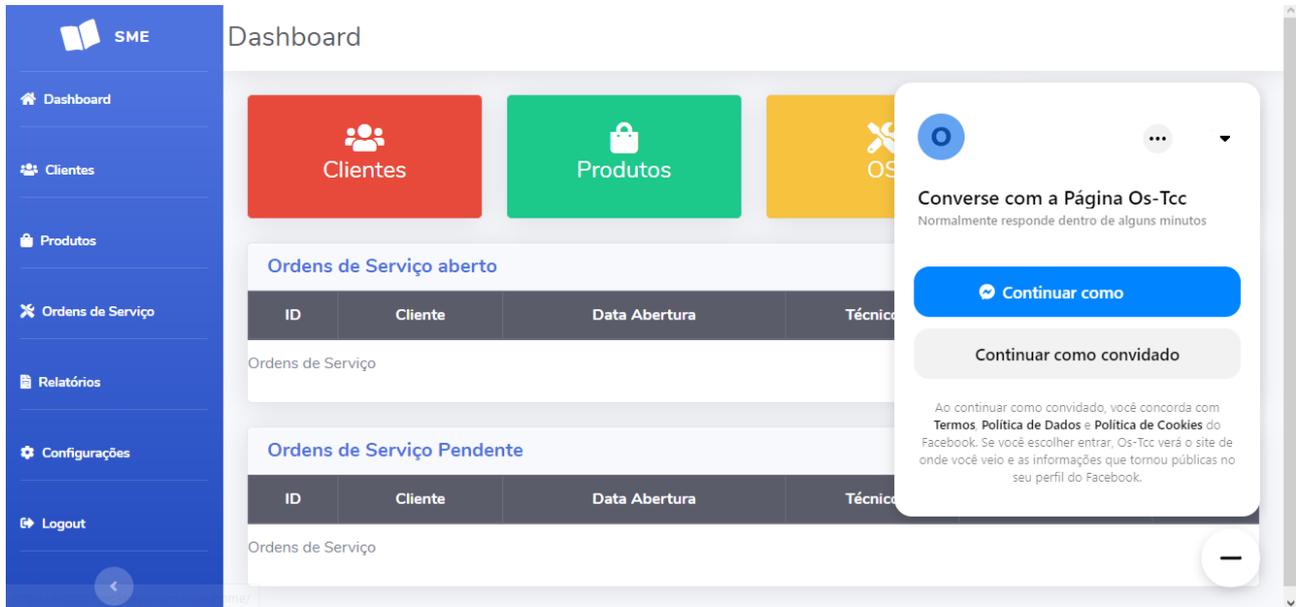


FIGURA 25: TELA INICIAL COM MESSENGER ABERTO

Fonte: O autor

5. Conclusão

5.1. RESULTADOS

A partir dos resultados adquiridos, é possível afirmar que, com o uso da Morgana, os usuários passam a contar com mais um canal específico para busca de conhecimento e ajuda, sem a necessidade de se ter uma pessoa fisicamente para esclarecimentos de dúvidas, contando assim como uma vantagem da implementação de sistemas de *chatbot*.

Dentre as desvantagens destaca-se o fato de que nem todas as questões levantadas pelo usuário foram respondidas devido a questões de entendimento, levando em conta que o *bot* não tem tamanha inteligência quanto o homem possui para contextos.

Para criar o diálogo que respondeu aos principais questionamentos foi necessário criar: 20 intenções e 15 entidades.

A utilização do Watson assistente possibilitou a criação do *chatbot* como uma ferramenta sem necessidade de codificação ou montagem de estruturas físicas já que todas as suas funcionalidades estão disponíveis nas nuvens gratuitamente, sendo este formato considerando o uso de até 10 mil chamadas API mês, limitando a quantidade de 5 áreas de trabalho, 25 entidades e 100 intenções no diálogo.

5.2. CONSIDERAÇÕES FINAIS

Com o desenvolvimento da tecnologia no campo da inteligência computacional e o desenvolvimento da pesquisa, os *chatbots* estão cada vez mais aparecendo no dia-a-dia das pessoas na forma de serviços, atendimento ao cliente, perguntas frequentes e formas ainda mais avançadas.

O trabalho visou a criação de um *chatbot* para auxiliar no atendimento em um sistema web, desta forma aumentar a velocidade da resolução de problemas e atendimento da equipe de suporte técnico. Uma das vantagens na utilização do *chatbot* é a capacidade responder as perguntas frequentes em um curto tempo, desta forma serviços triviais não será necessário o atendimento físico de um técnico e/ou deslocamento da equipe até o local e atendimento. Outra vantagem do *chatbot* é a capacidade dele de manter um diálogo simultâneo com diversas pessoas, diferente de um atendente físico.

No contexto de aprendizagem de máquina, quanto maior o número de interações que o *chatbot* tiver, melhor será seu desempenho em responder perguntas de seu domínio.

Por meio do apresentado anteriormente neste trabalho, pode-se afirmar que a solução apresentada consegue atender as necessidades dos usuários em relação aos problemas mais triviais e dúvidas frequentes. Caso não haja solução do problema, ainda aumenta a eficiência do atendimento direto ao suporte, devido filtrar os problemas que necessitam do auxílio da equipe e fornecendo uma visão ampla sobre o funcionamento do sistema. Pensando mais a longo prazo, a consulta aos diversos sistemas agrega valor ao serviço prestado dos equipamentos pré-especificados.

Como proposta de trabalhos futuros, a criação de um maior fluxo de diálogo com o usuário, envolvendo uma maior quantidade de assuntos e demonstrações de mídias para auxiliar no conteúdo faz-se pertinente. Deste modo, num momento posterior, o usuário poderá conduzir o diálogo por meio de comandos de voz; mantendo, assim, a linha de diálogo que foi apresentada neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

AHO, A. V.; ULLMAN, J. D. **The Theory of Parsing, Translation, and Compiling**, Vol. 1, Parsing. Prentice Hall, 1972.

Brito, Neves Felipe. **DESENVOLVIMENTO DE UM CHATTERBOT PARA A PÁGINA WEB DE UM CURSO DE NÍVEL SUPERIOR**. Universidade Federal do Ceará, Quixadá, 2017.

CHOMSKY, N. **Syntactic Structures**. The Hague: Mouton. Reprint. Berlin and New York, 1957.

CHOMSKY, N. **Aspects of the theory of syntax**. Cambridge, Mass.: MIT Press, 1965.

COLBY, K. M. **Simulation of Belief Systems. Computer Models of Thought and Language**. In: R. C. Schank and K. M. Colby. San Francisco, W.H. Freeman and Company: 251-286, 1973.

COLMERAUER, A. **Total Precedence Relations**. *Journal of the ACM*, 17(1): 14-30, 1970.

COLMERAUER, A. **Les Grammaires de Metamorphos, Technical Report, Groupe d'Intelligence Artificielle**, Marseille-Luminy, November, 1975.

COLMERAUER, A.; ROUSSEL, P. **The birth of Prolog History of Programming Languages**. ACM Press, 1996.

Comarella, Lunardi Rafaela; Café, A. Maria Ligia. **CHATTERBOT: conceito, características, tipologia e construção**. *Inf. & Soc: Est., João Pessoa*, v.18, n.2, p. 55-67, maio/ago. 2008.

FACEBOOK– CHAT PLUGIN – Disponível em: <
<https://developers.facebook.com/docs/messenger-platform/discovery/facebook-chat-plugin/> > acessado em: 22 de junho de 2020

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, 2010.

IBM – Watson – **Inteligência Artificial** – Disponível em: < <https://www.ibm.com/watson>>. Acesso em: 20 junho 2020.

IBM, < <https://cloud.ibm.com/apidocs/assistant/assistant-v2?code=python> > Acesso em 20 de junho de 2020

JURAFSJY, D.; MARTIN, J. **Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics**. Pearson Prentice Hall, 2009.

KAR, R.; HALDAR, R. **Applying chatbots to the internet of things: Opportunities and architectural elements**. arXiv preprint arXiv:1611.03799, 2016.

KUNZE, L. **What is a chatbot, how do you build one, who is building them and are they good?** [S.l.], 2016. Disponível em: < <https://www.techworld.com/apps-wearables/what-is-chatbot-how-do-you-build-one-who-is-building-them-are-they-any-good-3638740/>>. Acesso em: 13/03/2020.

PEREIRA, F. C. N. & WARREN, D. H. D. **Definite Clause Grammars for Language Analysis – A Survey of the Formalism and a Comparison with Augmented Transition Networks**. Artificial Intelligence, 13: 231-278, 1980.

PERNA, L. CRISTINA; DELGADO, K. HELOÍSA; FINATTO, J. MARIA. **Linguagens Especializadas em CORPORA. Modos de Dizer e Interfaces de Pesquisa**. EDIPUCS- Editora Universitária da Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2010.

RABINER, L. R., HUANG, B. H. An Introduction to Hidden Markov Models, IEEE ASSP Magazine, January 1986.

RICH, E.; KNIGHT, K. **Inteligência Artificial**. Makron Books, 1993.

ROSA, G. LUIS JOÃO. **O Significado da Palavra para o Processamento de Linguagem Natural**. Pontifícia Universidade Católica de Campinas, 1997.

Santos, Diana. **Introdução ao processamento de linguagem natural através das aplicações.** In Elisabete Ranchhod (ed.), Tratamento das Línguas por Computador. Uma introdução à linguística computacional e suas aplicações, Lisboa: Caminho, p. 229-259.

Silva, André Marcos & Mattos, Rogério. **IBM Watson como Ambiente para Desenvolvimento e Execução de um Chatbot – Um Estudo de Caso Aplicado ao Processo de Atendimento ao Usuário.** Centro Universitário Adventista de São Paulo (UNASP), 2018.

SILVA, Bento Carlos Dias da. **O estudo Lingüístico-Computacional da Linguagem.** Letras de Hoje, Porto Alegre, v. 41, n.2, p. 103-138, jun. 2006.

SILVA, M. B. M. FILIPA DANIELA. **Algoritmos de processamento de linguagem natural para sistemas de conversão texto-fala em português.** Dissertação submetida para obtenção da “mención de DOUTOR EUROPEO”, Faculdade de Filologia da Universidade da Coruña, 2008.

Turing, A.M.(1950) **Computer Machinery and Intelligence.** Mind. Vol. 9, No. 236, pp. 433-460.

WEIZENBAUM, JOSEPH. **A Computer Program for the Study of Natural Language Communication Between Man and Machine.** Massachusetts Institute of Technology, Volume 9, Number 1 (January 1966): 36-35.

WINOGRAD, T. **Procedures as a Representation for Data in a Computer program for Understanding Natural Language.** M.I.T. dissertation, 1971.

WOODS, W. A. **Lunar rocks in natural english: Explorations in natural language question answering.** In: A. Zampoli, editor, Linguistic Structures Processing, volume 5 of Fundamental Studies in Computer Science, 521-569. North-Holland, 1977.