



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Augusto Luiz Greuel

**Assistente virtual do tipo chatbot para gestão de eventos de uma infraestrutura
de TI**

Florianópolis
2020

Augusto Luiz Greuel

Assistente virtual do tipo chatbot para gestão de eventos de uma infraestrutura de TI

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.

Orientador: Prof. Eric Aislan Antonelo, Dr.
Supervisor: Gabriel Assis Amancio, Eng.

Florianópolis
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Greuel, Augusto

Assistente virtual do tipo chatbot para gestão de eventos de uma infraestrutura de TI / Augusto Greuel ; orientador, Eric Antonelo, 2020.

73 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2020.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Chatbot. 3. Assistente virtual. 4. Processamento de Linguagem Natural. I. Antonelo, Eric . II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Augusto Luiz Greuel

Assistente virtual do tipo chatbot para gestão de eventos de uma infraestrutura de TI

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Eric Aislan Antonelo, Dr.
Orientador
UFSC/CTC/DAS

Gabriel Assis Amancio, Eng.
Supervisor
Empresa WEG

Prof. Jomi Fred Hübner, Dr.
Avaliador
UFSC/CTC/DAS

Prof. Fabio Luis Baldissera, Dr.
Presidente da Banca
UFSC/CTC/DAS

AGRADECIMENTOS

À Universidade Federal de Santa Catarina, especialmente ao Departamento de Automação e Sistemas por todas as experiências e ensinamentos repassados ao longo dos anos.

Ao meu orientador, Prof. Dr. Eric Aislan Antonelo pelo suporte e direcionamento sugeridos durante o desenvolvimento deste projeto.

Ao meu supervisor, Eng. Gabriel Assis Amancio pela confiança e disponibilidade de executar estas e outras atividades dentro da empresa, tornando a experiência de estágio extremamente engrandecedora.

À equipe de segurança da informação da WEG, Sergio, Rejane, Ademar e Pierre que demonstraram paciência e flexibilidade ao me orientarem dentro desta área complexa na empresa.

À minha família e amigos, que sempre estiveram presentes para dar suporte durante todas as dificuldades e que também estiveram por perto nos momentos felizes.

*“É na distância, não na velocidade, que se encontram as respostas.
A recompensa vem quando você confronta e
ultrapassa os limites do cansaço.”
(Bingham, 2003)*

RESUMO

A demanda por acesso à informação de maneira rápida e eficiente é uma tendência global em empresas que buscam estar superando concorrentes e se destacando no mercado. Provendo maior autonomia aos seus funcionários, estas empresas podem vir a aumentar a sua produtividade geral, entregando resultados melhores e mais rápidos. O presente trabalho busca desenvolver um assistente virtual do tipo chatbot para simplificar o acesso à informação de uma plataforma de gestão de eventos de infraestrutura de TI de uma empresa, visando aumentar a autonomia dos seus colaboradores. Apresentamos a criação de uma arquitetura, desenvolvida através da linguagem Python, responsável por realizar o processamento de linguagem natural por meio da ferramenta Dialogflow e ainda converter a intenção do usuário em pesquisas dentro da plataforma de gerenciamento de eventos (SIEM) da empresa. Relatamos também a criação da interface gráfica do assistente virtual com suporte das linguagens HTML, CSS e Javascript para possibilitar a visualização das informações trocadas entre o usuário e o chatbot de maneira textual e gráfica. A avaliação das ferramentas implementadas é realizada por meio de uma pesquisa com os utilizadores da plataforma. Os resultados obtidos demonstram uma avaliação satisfatória dos recursos e também a possibilidade de extensão da solução para outras áreas da empresa.

Palavras-chave: Assistente Virtual. Chatbot. Python. Processamento de Linguagem Natural. Dialogflow. SIEM. HTML. CSS. Javascript.

ABSTRACT

The demand for access to information quickly and efficiently is a global trend in companies that seek to outperform competitors and stand out in the market. By providing greater autonomy to their employees, these companies can increase their overall productivity, delivering better and faster results. The present work seeks to develop a chatbot virtual assistant to simplify the access to information from a company's IT infrastructure event management platform, aiming to increase the autonomy of its employees. We present the creation of an architecture, developed through the Python language, responsible for performing natural language processing through the Dialogflow tool and also converting the user's intention into searches within the company's event management platform (SIEM). We also report the creation of the graphical interface of the virtual assistant with the support of HTML, CSS and Javascript to enable the visualization of the information exchanged between the user and the chatbot in a textual and graphical way. The assessment of the implemented tools is carried out through a survey with the platform users. The results obtained demonstrate a satisfactory assessment of the implemented resources and also show the possibility of extending the solution to other areas of the company.

Keywords: Virtual Assistant. Chatbot. Python. Natural Language Processing. Dialogflow. SIEM. HTML. CSS. Javascript.

LISTA DE FIGURAS

Figura 1 – Simulação de conversação com a chatbot ELIZA.	22
Figura 2 – Exemplo de configuração de arquivo AIML da chatbot ALICE.	23
Figura 3 – Simulação de conversação com a chatbot Mitsuku.	24
Figura 4 – Modelo <i>seq2seq</i> para geração das respostas do chatbot.	25
Figura 5 – Avaliação do Meena em comparação com humanos e outros chatbots	26
Figura 6 – Serviços da plataforma Azure para desenvolvimento de uma aplica- ção de chatbot.	29
Figura 7 – Diagrama da arquitetura do chatbot.	32
Figura 8 – Diagrama de casos de uso do assistente virtual.	35
Figura 9 – Exemplo de frases de treinamento para uma intenção de extração de informações do clima.	38
Figura 10 – Ação disparada no momento em que a intenção de extração de clima foi ativada.	39
Figura 11 – Exemplo de contextos de uma intenção.	40
Figura 12 – Exemplos de respostas de uma intenção.	40
Figura 13 – Exemplo de respostas de uma intenção <i>fallback</i>	41
Figura 14 – Seleção de entidades em frases de treinamento.	42
Figura 15 – Extração de parâmetros de uma frase de treinamento por meio do objeto JSON.	43
Figura 16 – Exemplo de uma pesquisa realizada no Splunk através da linguagem SPL	44
Figura 17 – Exemplo de código HTML para criação de uma página WEB.	46
Figura 18 – Exemplo de código CSS para estilização de uma página WEB.	47
Figura 19 – Interface de usuário do assistente virtual	49
Figura 20 – Interação com o assistente para extração de e-mails bloqueados.	50
Figura 21 – Resposta JSON do Dialogflow à primeira iteração do usuário.	51
Figura 22 – Resposta JSON do Dialogflow à segunda iteração do usuário.	52
Figura 23 – Resposta JSON do Dialogflow à terceira iteração do usuário	53
Figura 24 – Conversa exemplo para extração do perfil de acesso à internet.	54
Figura 25 – Conversa exemplo para extração dos e-mails bloqueados.	55
Figura 26 – Conversa exemplo para verificação de modificações em arquivo da rede.	56
Figura 27 – Conversa exemplo para extração de grupos do AD de um usuário.	58
Figura 28 – Interface inicial do assistente para gestores.	59
Figura 29 – Negação da extração de dados de outro colaborador	60
Figura 30 – Conversa exemplo para extração dos e-mails enviados por um cola- borador da equipe do gestor	61

Figura 31 – Respostas dos usuários à pergunta: "Como você avalia a maneira de se comunicar do assistente?"	64
Figura 32 – Respostas dos usuários à pergunta: "O assistente conseguiu otimizar o seu tempo através de alguma funcionalidade?"	65
Figura 33 – Respostas dos usuários à pergunta: "O assistente conseguiu simplificar o acesso à alguma informação que você procurava?"	66
Figura 34 – Respostas dos usuários à pergunta: "Como você avalia a interface do assistente e o formato de apresentação de informações (gráficos, tabelas e texto)?"	66

LISTA DE ABREVIATURAS E SIGLAS

AD	<i>Active Directory</i>
AIML	<i>Artificial Intelligence Markup Language</i>
ALICE	<i>Artificial Linguistic Internet Computer Entity</i>
API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheet</i>
HTML	<i>Hypertext Markup Language</i>
IA	Inteligência Artificial
JSON	<i>JavaScript Object Notation</i>
LSTM	<i>Long Short-Term Memory</i>
LUIS	<i>Language Understanding Intelligent Service</i>
PLN	Processamento de Linguagem Natural
SIEM	<i>Security Information and Event Management</i>
SPL	<i>Search Processing Language</i>
SQL	<i>Structured Query Language</i>
SSA	<i>Sensibleness and Specificity Average</i>
TI	Tecnologia da Informação

SUMÁRIO

1	INTRODUÇÃO	13
1.1	PROBLEMA	14
1.2	OBJETIVO GERAL	14
1.3	OBJETIVOS ESPECÍFICOS	15
1.4	METODOLOGIA	15
1.5	A EMPRESA	16
1.6	ORGANIZAÇÃO DO DOCUMENTO	17
2	REVISÃO DA LITERATURA	18
2.1	INTELIGÊNCIA ARTIFICIAL	18
2.2	PROCESSAMENTO DE LINGUAGEM NATURAL	19
2.3	CHATBOTS	20
2.3.1	Evolução dos chatbots	21
2.3.1.1	ELIZA	21
2.3.1.2	ALICE	22
2.3.1.3	Mitsuku	24
2.3.2	Redes neurais aplicadas aos chatbots atuais	25
2.3.3	Soluções de mercado para interpretação de linguagem natural	27
2.3.3.1	Dialogflow	27
2.3.3.2	Watson Assistant	28
2.3.3.3	LUIS	28
2.3.3.4	Lex	29
2.3.3.5	Wit.ai	30
3	PROPOSTA	31
3.1	ARQUITETURA DO CHATBOT	31
3.2	ESPECIFICAÇÕES DE REQUISITOS	33
3.2.1	Requisitos funcionais	33
3.2.2	Requisitos não funcionais	34
3.3	CASOS DE USO	34
3.3.1	Descrição dos casos de uso	35
4	FERRAMENTAS E TECNOLOGIAS	37
4.1	DIALOGFLOW	37
4.1.1	Intenções	37
4.1.1.1	Frases de treinamento	37
4.1.1.2	Ação	38
4.1.1.3	Contexto	39
4.1.1.4	Respostas	40
4.1.1.5	Intenções generalizadas	41

4.1.2	Entidades	41
4.2	SPLUNK	43
4.3	PYTHON	44
4.3.1	Flask	45
4.4	JAVASCRIPT, HTML E CSS	45
5	DESENVOLVIMENTO	48
5.1	INTERFACE DO USUÁRIO	48
5.2	PRINCÍPIO DE FUNCIONAMENTO	50
5.3	FUNCIONALIDADES	53
5.3.1	Extração de perfil de acesso à internet	53
5.3.2	E-mails bloqueados	54
5.3.3	Modificações de arquivos	56
5.3.4	Extração dos grupos do AD	57
5.3.5	Extrações de dados de outros colaboradores	58
5.3.6	E-mails trocados	60
6	ANÁLISE DE RESULTADOS	62
6.1	USO DAS FUNCIONALIDADES	62
6.2	AVALIAÇÃO DAS FUNCIONALIDADES	63
7	CONSIDERAÇÕES FINAIS E PERSPECTIVAS	67
7.1	SUGESTÕES DE TRABALHOS FUTUROS	68
	REFERÊNCIAS	69

1 INTRODUÇÃO

A tecnologia se tornou a principal ferramenta de comunicação em empresas e organizações. Ao longo dos anos ela alterou a forma com a qual as empresas interagem com o seu público e também como os seus próprios colaboradores se comunicam. As organizações que não priorizam o tema tendem a ser superadas por outras empresas que vêm incorporando mudanças e estudando novas soluções (UNIVERSITY, 2020).

O constante aprimoramento de máquinas e algoritmos que são capazes de aprender com base em conjuntos de dados introduz novas soluções para comunicação. Sistemas inteligentes podem solucionar problemas que antes eram considerados extremamente complexos, ampliando assim a gama de aplicações disponíveis (GUNKEL, 2017). O debate do tema fez com que pesquisadores iniciassem o desenvolvimento de soluções cada vez mais capazes, instigando perguntas como "As máquinas podem pensar?" (TURING, 1996). Com o surgimento da inteligência artificial (IA) começa-se a questionar a possibilidade de utilizar computadores para simular a inteligência humana através de características como a capacidade de entender, pensar e aprender (RAHIM *et al.*, 2018, p. 2). As tecnologias evoluíram e fizeram as empresas se atentarem ao tema.

A utilização da inteligência artificial em empresas traz a possibilidade de ganhar vantagens competitivas em relação aos concorrentes, agregando valor ao negócio. A IA vem possibilitando uma mudança na forma de comunicação entre as pessoas e as máquinas. Interfaces conversacionais são tratadas atualmente como mecanismos essenciais de interação com clientes. Além disso, através de assistentes virtuais, a IA possibilita a criação de mecanismos para ajudar pessoas em domínios extramente específicos que requerem algum tipo conhecimento prévio (DALE, 2016).

Os assistentes virtuais, também chamados de chatbots, estão sendo cada vez mais utilizados em diversas empresas como alternativas para aumentar a produtividade das suas atividades e facilitar o contato dos funcionários com plataformas complexas. Segundo analistas do Gartner, 70% dos trabalhadores com funções administrativas e/ou executivas estarão em contato diário com plataformas de conversação do tipo chatbot até 2022 (GOASDUFF, 2019).

A evolução das tecnologias que estão por trás dos chatbots fez com que eles se tornassem muito mais abrangentes, possibilitando conversas mais complexas e até a análise de contextos em conversações. As melhorias no processamento de linguagem natural (PLN) tornaram viáveis conversas em níveis quase humanos fazendo com que a interação entre o usuário e o chatbot seja muito mais natural. Os assistentes já são capazes de possuírem personalidades próprias criando identidades únicas e fazendo com que as empresas possam construir relações mais íntimas com seus clientes. Além

disso, o fato de poder utilizar um software para realizar grande parte do processo de interação com usuários faz com que seja possível aumentar o número de respostas à qualquer hora do dia, oferecendo um engajamento cada vez maior (VAISH, 2019).

1.1 PROBLEMA

Um SIEM (do inglês, *Security Information and Event Management*) é uma plataforma de armazenamento e gerenciamento de registros de eventos computacionais (*logs*). Através desta ferramenta é possível correlacionar dados de diversos tipos de aplicações de uma infraestrutura de TI de uma empresa e com isso gerar uma rede de rastreamento de atividades. Pode-se identificar anomalias na utilização de recursos de uma determinada aplicação, categorizar dados, gerar relatórios de incidentes e monitorar ambientes. Este tipo de solução suporta pesquisas e extrações de informações em volumes de dados extremamente grandes, tornando viável o monitoramento de diversos recursos de TI em uma corporação (PRATT, 2017).

Uma das soluções mais relevantes de SIEM disponíveis no mercado é conhecida como Splunk. O Splunk possui uma linguagem de programação própria, chamada de *Search Processing Language* (SPL). Esta linguagem foi desenvolvida para simplificar a extração de informações de arquivos de sistemas e programas formatados em eventos. Ela foi criada com base na linguagem de consulta SQL (do inglês, *Structured Query Language*), muito utilizada para a extração de informações de banco de dados relacionais. Por este motivo, a SPL possui características muito similares à SQL. Apesar disto, a SPL foi criada para otimizar a consulta a registros de eventos temporais de computadores, também conhecidos como *logs* (CARASSO, 2010). Por este motivo, a linguagem possui as suas próprias nuances.

O fato do Splunk possuir uma linguagem de programação exclusiva limita a utilização da ferramenta. É necessário que os usuários da plataforma possuam algum conhecimento prévio para poderem extrair informações úteis das suas consultas. Este tipo de pré-requisito gera um problema de acessibilidade à informação que está armazenada dentro plataforma. Cria-se então uma barreira de acessibilidade, fazendo com que seja necessário o treinamento de colaboradores específicos que possam ajudar os usuários a consultarem os dados que estão armazenados.

1.2 OBJETIVO GERAL

O principal objetivo deste projeto é democratizar o acesso dos dados que estão presentes no SIEM da empresa para que mais colaboradores possam extrair informações do software, sem a necessidade de conhecimento da linguagem de programação específica deste. Para isso será implementado um assistente virtual do tipo chatbot,

com o qual os usuários possam interagir de maneira simplificada através da troca de mensagens.

1.3 OBJETIVOS ESPECÍFICOS

Os objetivos específicos, que se referem aos pontos de estudo que serão levados em consideração durante este projeto, estão listados abaixo.

- Realizar pesquisas em materiais bibliográficos relacionados ao tema de processamento de linguagem natural e também desenvolvimento de assistentes virtuais do tipo chatbot.
- Avaliar as diferentes soluções de mercado que envolvem interpretação de linguagem natural e selecionar a opção mais adequada ao projeto.
- Levantar requisitos e definir funcionalidades para a criação do assistente virtual.
- Desenvolver o assistente e criar uma interface gráfica com a qual o usuário possa interagir.
- Garantir a restrição de acesso de informações de acordo com o perfil do usuário.
- Realizar testes para avaliar o desempenho e o funcionamento do chatbot.
- Analisar os resultados obtidos e avaliar as funcionalidades através de pesquisas com os usuários.
- Avaliar a utilização deste tipo de solução para outras necessidades da empresa.

1.4 METODOLOGIA

A estruturação do projeto deve ser adaptada para possibilitar o desenvolvimento por completo de uma aplicação do tipo software. Para isso, criou-se um planejamento de atividades que foi concebido com a utilização de ferramentas de gerenciamento de projetos na empresa. Com base no levantamento destas atividades, pode-se organizar prioridades para possibilitar o desenvolvimento de cada etapa do projeto.

A etapa inicial consiste no treinamento e conhecimento, por parte do autor, das tecnologias de processamento de linguagem natural. Após esta etapa, é necessário ponderar entre as diversas opções de mercado e escolher uma delas para a utilização no decorrer do desenvolvimento das atividades.

Em paralelo aos estudos dos temas do projeto, é preciso realizar uma pesquisa com os usuários chave para que seja possível identificar as principais demandas dos colaboradores com relação a problemas de acesso à informação disponíveis no SIEM

da empresa. Com base nestas pesquisas serão levantados os principais recursos da aplicação.

Após o levantamento das funcionalidades e a escolha dos recursos tecnológicos, inicia-se a etapa de desenvolvimento da aplicação. Nesta etapa, cria-se uma arquitetura que será responsável por tratar todo o funcionamento idealizado nas etapas de levantamento de requisitos. Esta arquitetura deve ser desenvolvida para minimizar o retrabalho de implementação de novas funcionalidades e também possibilitar ao usuário uma experiência de utilização simples e eficiente.

Por fim, deve-se realizar testes e rodadas de aperfeiçoamento com usuários específicos para tratar de problemas e melhorias que devem ser implementadas anteriormente ao processo de disponibilização para o público geral do projeto.

Todas as atividades aqui discutidas e executadas são acompanhadas por um sistema de gerenciamento de projetos fornecido pela empresa. As entregas são concebidas por meio de uma metodologia ágil para desenvolvimento de software chamada de SCRUM. A metodologia consiste na subdivisão das entregas em pequenos espaços de tempo conhecidos como *sprints*. Neste projeto cada *sprint* possui duração de 15 dias. No início de cada semana é feita uma reunião com a gerente de projetos para estabelecer as principais entregas daquela *sprint* e também discutir eventuais problemas e melhorias.

1.5 A EMPRESA

A WEG é uma empresa com sede na cidade de Jaraguá do Sul em Santa Catarina, criada no ano de 1961. É considerada uma das maiores fabricantes de equipamentos elétricos do mundo, possuindo indústrias em 12 países e contando com mais de 31 mil colaboradores. A área de atuação é ampla e a empresa pode ser dividida em cinco principais negócios: motores elétricos, automação, transmissão e distribuição, geração de potência e ainda tintas e vernizes (WEG, 2020).

Está presente em diversos segmentos da indústria contando com soluções para empresas do ramo de petróleo e gás, mineração, papel e celulose, energias renováveis, açúcar, etanol, geração de energia, agronegócio, mobilidade elétrica, aço, transmissão e distribuição, entre outros. Todas essas vertentes demonstram a forte capacidade de produção da empresa, se destacando globalmente como fornecedora de soluções.

Mais recentemente a WEG iniciou a criação de produtos envolvendo soluções digitais. Aplicando conceitos abordados na Indústria 4.0, vem buscando trazer ao mercado recursos envolvendo Inteligência Artificial, *Big Data* e Internet das Coisas para conectar processos produtivos e gerar dados de análise. Toda essa conectividade corrobora para que as empresas possam tomar decisões com maior assertividade e rapidez, através da implantação de novos produtos que estão sendo desenvolvidos pela WEG (FUCUCHIMA, 2020).

1.6 ORGANIZAÇÃO DO DOCUMENTO

Este documento é organizado em sete capítulos. O primeiro capítulo destaca a metodologia, os problemas de estudo e também quais são os objetivos que o projeto busca atingir para solucioná-los.

No segundo capítulo é feita uma análise dos temas abordados no projeto com o objetivo de familiarizar o leitor com as nomenclaturas e áreas de estudos envolvendo chatbots. São abordados alguns dos principais artigos e livros relacionados ao tema, visando discutir as principais tecnologias que estão sendo aplicadas nos dias atuais e também algumas das tendências futuras na área.

No terceiro capítulo discute-se a arquitetura proposta para solucionar o problema de estudo, além de levantar os requisitos técnicos que deverão ser atendidos pela ferramenta em questão ao ser implementada.

No capítulo quatro realiza-se um análise das ferramentas utilizadas para a criação do projeto dando ênfase nos seus funcionamentos e também nas possibilidades de uso de cada uma delas.

No quinto capítulo descreve-se de maneira breve todo o processo de implementação de cada uma das funcionalidades do assistente virtual. Além disso, demonstra-se o esquema de funcionamento detalhado da arquitetura para um caso de uso.

No capítulo número seis é feita uma análise dos resultados obtidos por meio de uma pesquisa de utilização respondida por usuários do chatbot. Já no capítulo sete são elucidadas as considerações do autor sobre o projeto e discute-se também eventuais sugestões de trabalhos futuros.

2 REVISÃO DA LITERATURA

Neste capítulo são tratadas pesquisas e materiais bibliográficos que possuem relação com o tema desta monografia, com o objetivo de esclarecer e também gerar familiaridade com o assunto. São discutidos materiais sobre inteligência artificial, processamento de linguagem natural, chatbots e redes neurais.

2.1 INTELIGÊNCIA ARTIFICIAL

A inteligência artificial (IA) se baseia na tentativa de desenvolver comportamentos inteligentes para dispositivos. Dentre estes comportamentos estão envolvidas características como percepção, raciocínio, aprendizado, comunicação e atuação em ambiente complexos. Um dos principais objetivos da IA está no desenvolvimento de máquinas que podem realizar tarefas que envolvam estas características de maneira mais eficiente que os humanos (NILSSON; NILSSON, 1998).

Um dos primeiros eventos relatados envolvendo o tema de IA ao redor do mundo foi iniciado por volta do ano 1956. John McCarthy, um professor da universidade de Dartmouth, estava desapontado, porque um dos seus artigos recentes chamado de *Automata Studies* não continha informações relevantes relacionadas ao tema inteligência computacional. Neste mesmo ano, em um evento na universidade onde lecionava, McCarthy deu início a uma orientação de estudos para o entendimento e desenvolvimento da inteligência de máquinas, a qual ele intitulou como “Inteligência Artificial” (MOOR, 2006).

A IA se difere da computação convencional principalmente por lidar com aspectos de conhecimento e não somente com dados e informações. Ao longo dos anos de estudo foram desenvolvidas diferentes abordagens para representar o conhecimento nestas aplicações. Alguns exemplos amplamente conhecidos são formalismos baseados em lógica, sistemas de produção baseados em regras e representações procedurais (REITMAN, 1984).

A visão voltada à engenharia, do tema de inteligência artificial, trabalha principalmente com conceitos, teorias e práticas da criação de máquinas inteligentes. Algumas das aplicações que são encontradas neste ramo são máquinas que aconselham pessoas sobre assuntos específicos como finanças e até medicina, sistemas do tipo pergunta e resposta que conseguem interpretar diversos idiomas e também sistemas que realizam checagens técnicas em outros dispositivos para a verificação de pré-requisitos. Ainda na área da engenharia, são desenvolvidos robôs capazes de se comunicar com humanos através da conversação em linguagem natural e máquinas que podem realizar muitos outros dos chamados “trabalhos inteligentes” ao redor do mundo (GENESERETH; NILSSON, 2012).

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

O processamento de linguagem natural (PLN) é a área de estudos da inteligência artificial que busca entender e processar, por meio de computadores, a linguagem humana através da fala ou texto. O principal objetivo do PLN está em representar o significado e a intenção por trás de uma frase em linguagem natural por meio de técnicas de computação (LIDDY, 2001).

No ano de 1957 já são encontrados os primeiros relatos de estudos de sintaxe gramatical através de computadores. Apenas nas décadas de 70 e 80 que as pesquisas começaram a alavancar. Foram iniciados estudos envolvendo não somente a sintaxe, mas também a semântica, com o objetivo de gerar conceitos para compreender o significado das palavras. Nesta mesma época, a área de PLN começou a se consolidar, devido a aplicação de teorias linguísticas e da formalização de fatores pragmáticos e discursivos (TRAMUNT IBAÑOS; BATISTA PAIL, 2017).

A base de conhecimento para o estudo do processamento de linguagem natural aborda uma diversa gama de disciplinas. Estão envolvidos conceitos de ciência da computação, matemática, elétrica, eletrônica, robótica e psicologia. As soluções desenvolvidas podem ser aplicadas em áreas como tradução de idiomas, sumarização de conteúdo e reconhecimento de fala (CHOWDHURY, 2003).

Atualmente os métodos mais utilizados para o desenvolvimento de mecanismos de interpretação de linguagem natural continuam sendo baseados em sintaxe. Estes métodos podem ser sub-divididos em três principais grupos: identificação de palavras chave, afinidade lexical e métodos estatísticos. O primeiro método se baseia na classificação do texto por meio de palavras com o menor nível de ambiguidade na frase. Pelo fato de ser o mais simples de ser implementado, é também um dos mais comuns. O ponto fraco está na confiança da existência de palavras chave na frase que indicam o contexto ao qual ela pertence (CAMBRIA; WHITE, 2014).

O método de afinidade lexical adiciona uma camada de complexidade ao método de identificação de palavras chaves ao implementar um mecanismo de identificação de afinidade de certas palavras a determinadas categorias. Uma frase que contenha a palavra “acidente” tem maior afinidade a algo ruim e, portanto, o contexto desta frase tende a estar ligado a um evento negativo. O ponto fraco deste método fica evidente em frases que utilizam as palavras principais de afinidade em outros contextos, como por exemplo: “Eu a conheci por acidente”. Neste caso a palavra não está ligada a um termo negativo e, devido ao contexto, altera o significado da frase (CAMBRIA; WHITE, 2014).

Os métodos estatísticos, por sua vez, são os mais estudados nos dias atuais. Estes métodos se baseiam na utilização de algoritmos de aprendizado de máquina para tentar compreender o significado das frases. Estes algoritmos são treinados através de conjuntos com milhões de frases e após etapas de otimização passam

a ser capazes de interpretar frases nunca vistas anteriormente. Os resultados se destacam por serem capazes de compreenderem conceitos dos dois métodos citados anteriormente e ainda obterem maior acurácia. Entretanto, os métodos estatísticos têm como ponto negativo a necessidade de frases longas como parâmetro de entrada, devido ao mecanismo de relacionamento entre palavras que serve para a identificação de contextos (CAMBRIA; WHITE, 2014).

2.3 CHATBOTS

Os chatbots são aplicações de software que permitem a interação com usuário através da troca de mensagem em linguagem natural. Os métodos de comunicação são, na maioria dos casos, provenientes de mensagens de texto, entretanto em algumas aplicações podem ser utilizados áudios (DALE, 2016).

Desde o início dos estudos de desenvolvimento de chatbots muitos termos foram sendo designados para este tipo de ferramenta. Agentes virtuais, sistemas de diálogo e *chatterbot* são algumas das formas que essas aplicações foram chamadas ao longo do tempo. A principal característica destas soluções é simular uma conversação humana, emulando um diálogo comum, com o objetivo de solucionar algum problema específico (SHAWAR; ATWELL, 2007).

Pesquisadores realizaram um questionário para tentar entender a motivação dos usuários recorrentes de chatbot em diversas áreas de aplicação. Algumas das motivações encontradas estão relacionadas a entretenimento, fatores de relacionamento e também curiosidade. Entretanto, a principal motivação relatada pelos usuários é a produtividade. Os utilizadores relatam que a agilidade, facilidade e conveniência de utilizar um chatbot é o maior motivador para o uso da tecnologia. Além disso, essas pessoas destacam que os chatbots permitem assistência e acesso à informação. (BRANDTZAEG; FØLSTAD, 2017)

Ainda neste mesmo estudo, são destacadas algumas das características que levam os usuários a continuarem utilizando, ao longo do tempo, as soluções criadas por meio de chatbots. Os utilizadores relatam que os assistentes devem resolver algum problema ou alcançar um objetivo de maneira efetiva e eficiente. Eles devem agregar valor ao usuário aumentando a produtividade e garantindo acesso à informação. Além destes fatores, diversos usuários relatam que o entretenimento e as motivações sociais são aspectos importantes na experiência de utilização. Uma conversação em tom amigável e divertido motiva os utilizadores a voltarem a acessar a ferramenta (BRANDTZAEG; FØLSTAD, 2017).

2.3.1 Evolução dos chatbots

Alguns projetos de pesquisa envolvendo a criação de chatbots se tornaram destaques nesta área de pesquisa ao longo dos anos. Estas soluções geraram inovações nas abordagens de interpretação de linguagem natural e moldaram as tendências das tecnologias atualmente utilizadas.

2.3.1.1 ELIZA

Apesar do contínuo desenvolvimento de pesquisas e das otimizações dos algoritmos nos dias atuais, os chatbots já são conhecidos desde a década de 60. Neste período um pesquisador do MIT (*Massachusetts Institute of Technology*) desenvolveu a ELIZA, um programa de computador criado para possibilitar conversas em linguagem natural entre um homem e uma máquina (WEIZENBAUM, 1966).

O funcionamento da ELIZA é baseado na inspeção do texto para a identificação de palavras-chave. Quando uma palavra-chave é encontrada, a sentença é transformada com base em uma regra de associação para aquela palavra e então uma resposta é criada. As palavras-chave possuem um nível de precedência que é ranqueado, assim, quando uma sentença possui mais de uma destas palavras, a escolha é feita de acordo com o nível do ranque (WEIZENBAUM, 1966).

Os pilares de desenvolvimento técnico da ELIZA são baseados em 5 pontos principais (WEIZENBAUM, 1966):

1. Identificação da palavra-chave mais importante da mensagem do usuário.
2. Identificação do contexto no qual a palavra-chave pode aparecer.
3. Escolha e realização da regra de transformação para a palavra-chave elegida.
4. Criação de um mecanismo que permita a geração de respostas quando não forem identificadas palavras-chave na frase inserida pelo usuário.
5. Mecanismo que permita com facilidade a edição do código fonte da ELIZA.

Apesar da simplicidade do esquema de implementação da ELIZA, a chatbot obteve resultados relevantes. Na figura 1, pode-se notar uma conversa exemplo com o software sobre um tema relativamente complexo. Analisando as respostas da chatbot verifica-se que o entendimento dos contextos das mensagens do usuário funciona de maneira adequada através do esquema de identificação de palavras-chave. A conversa tende a ter continuidade, já que as respostas da ELIZA continuam instigando o usuário a falar cada vez mais sobre o seu problema.

Figura 1 – Simulação de conversação com a chatbot ELIZA.

```
User: Men are all alike.
ELIZA: IN WHAT WAY
User: They're always bugging us about something or other.
ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE
User: Well, my boyfriend made me come here.
ELIZA: YOUR BOYFRIEND MADE YOU COME HERE
User: He says I'm depressed much of the time.
ELIZA: I AM SORRY TO HEAR YOU ARE DEPRESSED
User: It's true. I am unhappy.
ELIZA: DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE
UNHAPPY
User: I need some help, that much seems certain.
ELIZA: WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
User: Perhaps I could learn to get along with my mother
```

Fonte – Adaptado de Weizenbaum

2.3.1.2 ALICE

ALICE (do inglês, *Artificial Linguistic Internet Computer Entity*) é um chatbot de código aberto desenvolvido no ano de 1995 pelo professor da universidade de Carnegie Mellon, Richard Wallace. Veio a ganhar popularidade por ter vencido o *Loebner Prize* por três anos (BRANDTZAEG; FØLSTAD, 2017). O prêmio *Loebner Prize* é uma competição anual que visa testar programas de computadores para tentarem simular o comportamento humano. Os jurados realizam conversações por meio de mensagens de texto e tentam descobrir se estão conversando com uma máquina ou com uma pessoa. O programa de computador que for melhor avaliado ganha a competição e recebe o prêmio.

A ALICE se caracteriza por ter uma estrutura modular que permite a inserção de novos modelos de interpretação de linguagem natural de maneira simplificada. O funcionamento se baseia no mapeamento, por meio de regras, de padrões de entrada do usuário com modelos uniformizados de saída. A ALICE possui um repositório extremamente grande de regras simples que foram implementadas ao longo do tempo. Ao invés de estruturas mais complexas como bases semânticas, morfológicas e sintáticas de processamento de linguagem natural, o funcionamento da ALICE tem como princípio o tamanho da sua base de dados de treinamento. As versões mais simples possuem cerca de cinquenta mil categorias de mapeamentos, entretanto, este número pode chegar na casa de milhões de categorias nos modelos mais complexos. (ABUSHAWAR; ATWELL, 2015)

As bases de conhecimento da ALICE ficam armazenadas em arquivos AIML (do inglês, *Artificial Intelligence Markup Language*). Estes arquivos foram criados pelos

próprios desenvolvedores do chatbot e permitem que outros programadores insiram diálogos de treinamento na ferramenta, facilitando o processo de aprendizado. Cada objeto AIML possui os chamados tópicos, que por sua vez, possuem as suas categorias. Cada categoria é uma regra que consiste na identificação de uma entrada e na conversão para uma determinada saída. No caso da ALICE existem três tipos de categorias possíveis: atômica, padrão e recursiva. A categoria atômica possui uma mensagem de texto fixa e uma resposta para esta mesma mensagem. Desta maneira, caso o usuário digite a mensagem desta categoria, ele receberá a resposta pré-programada. A categoria padrão possui um trecho de texto e um elemento curinga dentro da sua especificação. Neste caso, o usuário pode digitar uma mensagem que contenha aquele trecho e em seguida qualquer outro elemento e mesmo assim terá a resposta mapeada. Já a categoria recursiva permite aos programadores mapearem sinônimos a categorias de outros tipos. É possível criar diversas regras recursivas que identificam o mesmo resultado (ABUSHAWAR; ATWELL, 2015).

Um exemplo envolvendo a aplicação destas categorias em um arquivo AIML pode ser visto na figura 2. Em (1) nota-se uma categoria atômica, onde existe uma entrada (“*pattern*”) mapeada diretamente para uma saída (“*template*”). Em (2) cria-se uma categoria do tipo padrão, onde, devido ao elemento curinga, todas as frases que iniciarem pelo número 10 serão mapeadas para a resposta configurada na *tag* de *template*. Já em (3) pode-se verificar uma categoria recursiva, na qual a resposta está aninhada (*tag* “*<srai>*”) através de uma configuração feita para outra categoria com uma pergunta de mesmo objetivo. Neste caso existe um reaproveitamento de código para diminuir o retrabalho (ABUSHAWAR; ATWELL, 2015).

Figura 2 – Exemplo de configuração de arquivo AIML da chatbot ALICE.

```
1 <category>
  <pattern>10 Dollars</pattern>
  <template>Wow, that is cheap. </template>
</category>

2 <category>
  <pattern>10 *</pattern>
  <template>It is ten.</template>
</category>

3 <category>
  <pattern>DO YOU KNOW WHAT THE * IS</pattern>
  <template> <srai>what is <star/></srai> </template>
</category>
```

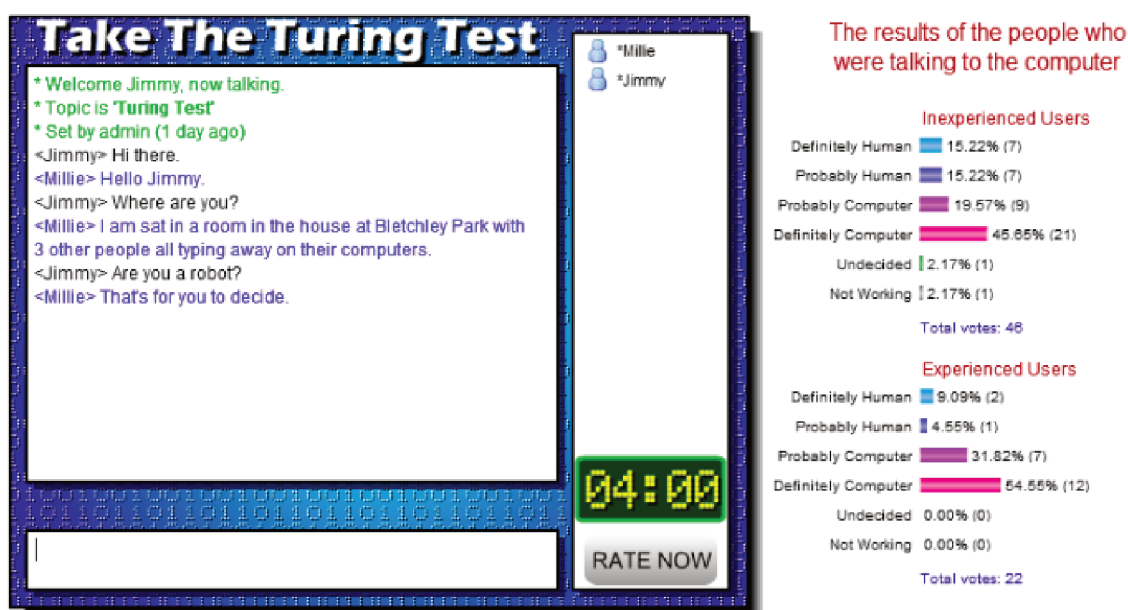

2.3.1.3 Mitsuku

A Mitsuku é atualmente considerada uma das melhores soluções de chatbot no quesito de interação com humanos. Vencedora de cinco prêmios *Loebner*, ela foi desenvolvida por Steve Worswick no ano de 2000. Assim como a ALICE, a Mitsuku utiliza arquivos do padrão AIML para a criação de regras de conversação (HIGASHINAKA *et al.*, 2014).

Uma das características desta chatbot está nas respostas para as perguntas com as quais ela não foi programada. Ironias e brincadeiras fazem parte das frases da Mitsuku para tentar simular o comportamento humano quando se depara com uma situação inusitada. Essa forma de implementação resolve alguns dos problemas encontrados na programação de chatbots por meio de arquivos AIML (WEERTS, 2018). Características como essa levaram a Mitsuku a ser considerada a melhor chatbot de conversação do mundo (ADIWARDANA *et al.*, 2020).

A figura 3 demonstra a versão da Mitsuku para a participação do *Loebner Prize* no ano de 2018. Durante a competição os chatbots são testados por usuários que possuem conhecimento na área de desenvolvimento deste tipo de aplicação (usuários experientes) e também por pessoas sem nenhum tipo de conhecimento prévio do tema (usuários não experientes). Destaca-se que mais de 30% do usuários não experientes que testaram a chatbot neste ano afirmaram que estavam conversando com um humano. Para os usuários experientes, o número ficou em aproximadamente 13%. Estes valores levaram a Mitsuku a vencer o prêmio daquele ano. (WORSWICK, 2018).

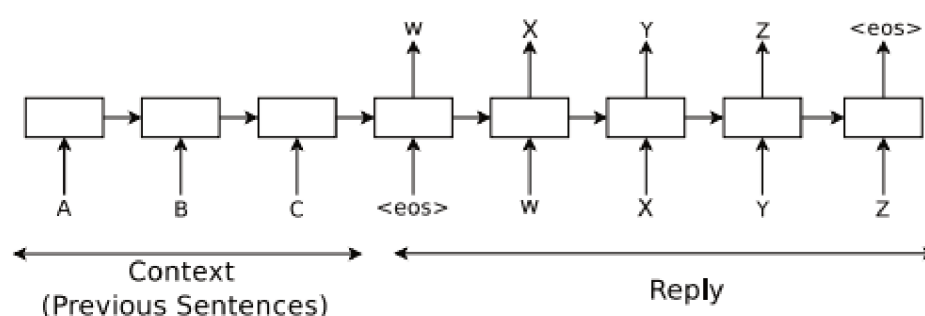
Figura 3 – Simulação de conversação com a chatbot Mitsuku.



2.3.2 Redes neurais aplicadas aos chatbots atuais

A maior parte das soluções que são atualmente referências em processamento de linguagem natural são baseadas em redes neurais recorrentes. As pesquisas sugerem a utilização destas redes para aplicações complexas, além de mecanismos mais tradicionais deste tipo de solução como por exemplo a classificação. Elas podem ser utilizadas para criação de um mapeamento de estruturas a outras estruturas. Através de modelos conhecidos como *seq2seq*, cientistas criaram chatbots que podem ser treinados através de longos diálogos, com pouca necessidade de aplicação de regras manuais. Estes modelos têm a capacidade de utilizar uma sequência de um diálogo como uma entrada e mapeá-la à uma outra sequência correspondente à resposta do chatbot, conforme a figura 4. Um codificador é responsável por processar as mensagens anteriores e um decodificador gera uma resposta. Com a utilização de redes neurais recorrentes do tipo *Long Short-Term Memory* (LSTM) pesquisadores puderam desenvolver um chatbot que consegue lembrar de conversas, entender contextos e criar respostas racionais sem a necessidade de criação de grandes conjuntos de regras manuais. O modelo ainda consegue generalizar novas respostas, não ficando restrito às frases utilizadas nas conversas de base de treinamento das redes neurais. Entretanto, existem avanços a serem feitos, já que o modelo ainda tende a responder com frases curtas e não aparentar possuir uma personalidade própria. Em casos onde a mesma pergunta foi realizada mais de uma vez, o chatbot respondeu com frases completamente diferentes (VINYALS; LE, 2015).

Figura 4 – Modelo *seq2seq* para geração das respostas do chatbot.

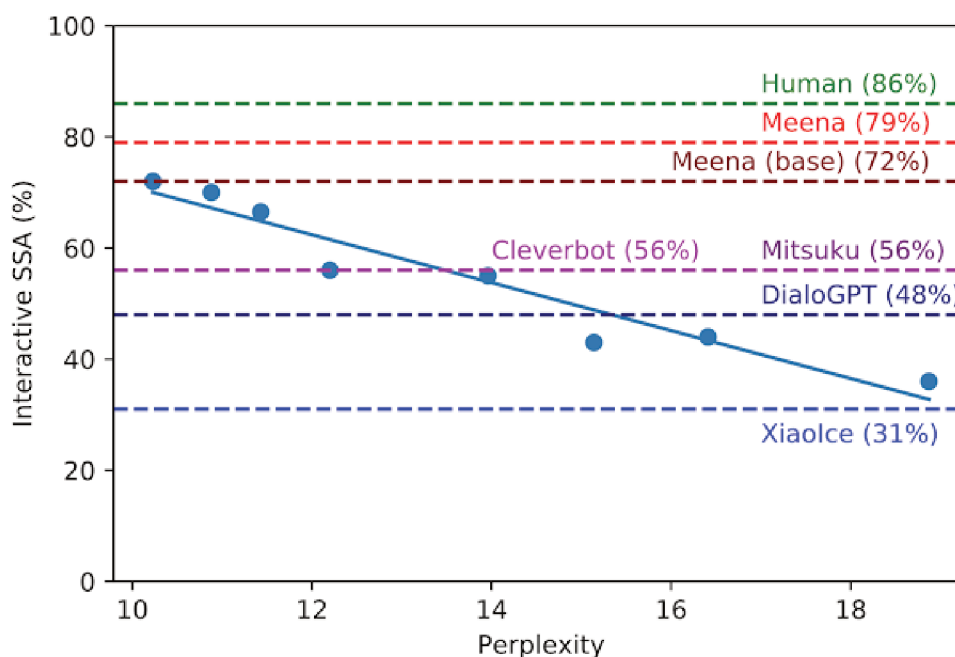


Fonte – Adaptado de Vinyals

No momento atual os chatbots mais complexos passaram a utilizar arquiteturas *Transformer*, ao invés de LSTM. Estas arquiteturas possuem camadas convolucionais em suas estruturas fazendo com que possam ser utilizadas tanto no codificador, quanto no decodificador do modelo *seq2seq*. Os resultados dessas pesquisas relatam modelos de chatbot como o Meena, que foi capaz de obter um índice de *Sensibleness and Specificity Average* (SSA) de 79%, enquanto humanos obtiveram o mesmo índice com

valor de 86%. Esta forma de avaliação de desempenho de chatbots representa uma média entre a sensibilidade e a especificidade de uma resposta em um diálogo, sendo avaliada por humanos. A sensibilidade relata se a resposta faz sentido, enquanto a especificidade relata se a resposta está dentro do contexto da conversa. Em outro formato de avaliação, o Meena obteve um índice de perplexidade de 10.2 em um dos seus modelos de treinamentos. Este índice é muito utilizado para testar modelos de probabilidade e ele prevê o quão bem um modelo consegue prever uma amostra. Na figura 5, pode-se notar, através das linhas tracejadas, a avaliação do SSA do Meena quando comparado com outras referências no mundo de chatbots e também com humanos. Além disso, verifica-se por meio dos pontos azuis, os índices de perplexidades obtidos em vários dos modelos de treinamento utilizados pelos autores. A linha de regressão, em destaque entre os pontos azuis, mostra que um alto SSA tem uma grande correlação com uma baixa perplexidade (ADIWARDANA *et al.*, 2020).

Figura 5 – Avaliação do Meena em comparação com humanos e outros chatbots



Fonte – Adaptado de Adiwardana

As redes neurais recorrentes com modelos *Transformer seq2seq* se demonstram referências nas soluções de chatbot mais complexas que foram produzidas até os dias atuais. Pesquisadores do Facebook relatam os seus avanços na área, através da produção de um chatbot que pode conversar sobre qualquer assunto. Através da aplicação de conjuntos de treinamento como publicações em redes sociais, bancos de dados contendo milhares conversas entre pessoas e também informações contidas em sites de pesquisas, os pesquisadores desenvolveram um chatbot que pode

se comunicar sobre praticamente qualquer tema. Apesar dos relatos de conversas sobre temas complexos, geração de engajamento e até um quesito de humanidade, os cientistas acreditam que ainda não chegaram em uma solução completamente robusta que possa discutir qualquer assunto. Os pesquisadores relataram que em algumas ocasiões o chatbot pode ser contradizer ou repetir as mesmas frases em conversas diferentes. O estudo deixa claro que ainda há espaço para avanços na área que continua evoluindo constantemente (ROLLER *et al.*, 2020).

2.3.3 Soluções de mercado para interpretação de linguagem natural

Com a crescente demanda para desenvolvimento de chatbots por parte de empresas para prover funcionalidades a clientes e também funcionários, diversas soluções de interpretação de linguagem natural foram sendo criadas. Estas soluções permitem que programadores criem modelos de treinamento para aplicações de chatbot específicas para as suas empresas, sem a necessidade de implementação do mecanismo de PLN.

2.3.3.1 Dialogflow

O Dialogflow é uma plataforma da Google que permite a formação de modelos de conversação via texto e voz para a criação de aplicações de interfaces conversacionais ou chatbots. Através da utilização de algoritmos de aprendizado de máquinas, a ferramenta permite que o programador tenha acesso ao mecanismo de processamento de linguagem natural para a criação do seu próprio assistente (GOOGLE, 2020).

Os desenvolvedores podem utilizar a plataforma WEB para criação de conversas modelo e treinarem os seus agentes. Cada formulário criado representa uma possibilidade de conversa com o chatbot e é configurado através de eventuais perguntas e as suas respectivas respostas. As respostas podem ser inseridas diretamente na plataforma ou podem ser extraídas por meio de outros mecanismos externos através da API (do inglês, *Application Programming Interface*) do software (CANONICO; DE RUSSIS, 2018).

Atualmente a plataforma permite a utilização dos seus recursos em mais de 30 idiomas diferentes. Além de contar com suporte técnico para diversas linguagens de programação como Python, C, Java, Go e PHP. Também possui um mecanismo de integração que possibilita de maneira simples a conexão do agente desenvolvido na plataforma com outras aplicações de chat, como por exemplo: Google Assistente, Skype, Telegram, Hangouts Chat e Facebook Messenger.

2.3.3.2 Watson Assistant

O Watson Assistant faz parte do pacote de inteligência artificial da IBM. Pode ser utilizado como um assistente virtual que é publicado em uma das plataformas de integração disponíveis como o Facebook Messenger ou o Slack. Além disso, é possível utilizá-lo como um recurso de desenvolvimento de aplicações próprias, permitindo a sua utilização em diversas outras áreas.

O funcionamento da ferramenta se baseia em um mecanismo de qualificação de diálogos. A mensagem do usuário é interpretada pela ferramenta e então é direcionada para um fluxo de conversa pré-programado pelo desenvolvedor. As perguntas que não foram implementadas e, portanto, não possuem respostas diretas, são mapeadas à uma base de dados com informações de conversas de assistentes desenvolvidos por meio da ferramenta ao redor do mundo. Desta forma, permite a interação com o usuário de maneira mais completa e eficiente (IBM, 2020).

O processo de treinamento para desenvolver o assistente consiste em três principais categorias: intenções, diálogos e entidades. As intenções são os objetivos do usuário ao interagir com o assistente. Portanto, todas as funcionalidades devem representar ao menos uma intenção, indicando quais são as tarefas que o usuário pode solicitar ao chatbot. Além das intenções que o usuário pode programar, existem diversas outras que já foram implementadas de maneira genérica. Caso o programador queira utilizá-las, ele pode realizar a ativação através de um menu de intenções. Os diálogos, por sua vez, representam os fluxos de conversação que acontecem quando o usuário interage com o assistente. Estes, são modelados através de uma estrutura gráfica, no formato de árvore, onde as mensagens são inseridas em níveis, ou ramificações da árvore. A última categoria são as entidades. Elas representam objetos que fornecem contexto a uma intenção. Exemplos de entidades são: datas, cidades, países e cores. Ou seja, são parâmetros que dentro de uma frase, ou intenção, ajudam a restringir o escopo de dados que eventualmente venham a ser apresentados a um usuário (IBM, 2020).

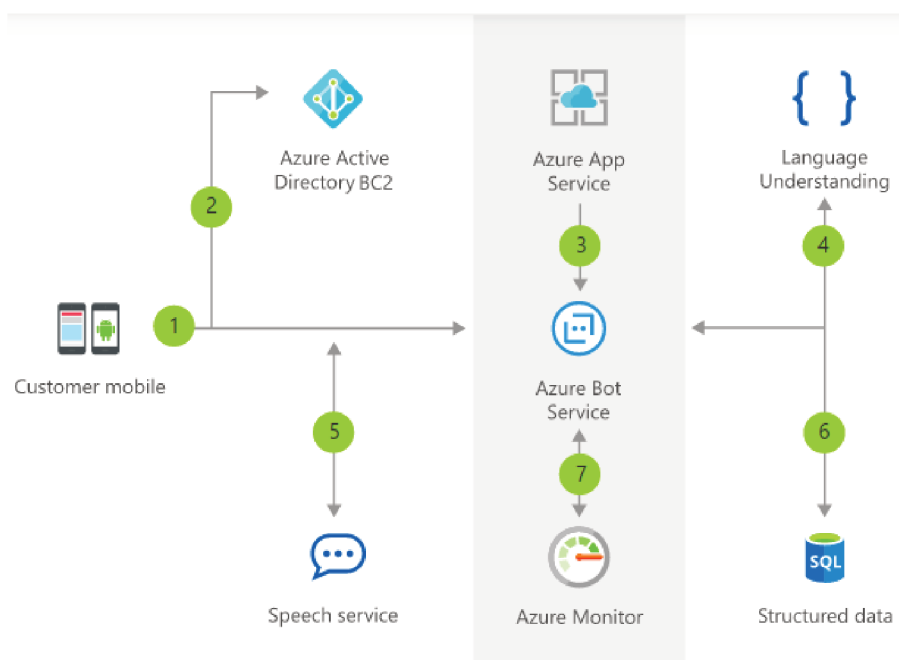
2.3.3.3 LUIS

O *Language Understanding Intelligent Service* (LUIS) é a plataforma de desenvolvimento de aplicações com processamento de linguagem natural da Microsoft. Permite a integração dos assistentes desenvolvidos em diversas plataformas como o e-mail do Office 365, Skype, Slack, Telegram, SMS e GroupMe. Utiliza tecnologias de aprendizado de máquina para interpretar os objetivos do usuário ao interagir com o chatbot e permitir a extração de informações chave daquelas interações. Possibilita o desenvolvimento de aplicações conversacionais inteligentes por parte de programadores, sem a necessidade de conhecimentos prévios em aprendizado de máquina

(MICROSOFT, 2020).

Pela possibilidade de integração com o Azure, que possui diversos pacotes de serviços criados pela Microsoft, permite a criação de aplicações complexas dispensando a agregação de ferramentas de outros fabricantes. Na figura 6, nota-se uma aplicação de atendimento ao cliente totalmente implementada através dos serviços Azure. O cliente acessa o assistente por meio do seu celular (1) e depois realiza a autenticação (2). O usuário faz uma solicitação (3) e a sua mensagem é processada pelo mecanismo de PLN (4). Uma resposta é enviada ao cliente que pode revisá-la para conferir as informações (5). Por fim, a demanda do usuário é efetuada (6) e os dados daquela solicitação são armazenados para posterior análise de desempenho (7) (MICROSOFT, 2020).

Figura 6 – Serviços da plataforma Azure para desenvolvimento de uma aplicação de chatbot.



Fonte – Adaptado de Microsoft

2.3.3.4 Lex

A solução de interpretação de linguagem natural da Amazon é conhecida como Lex. Suporta conexões diretas com o Facebook Messenger, Slack e Twilio. Além disso, possui integração simplificada para grande parte da gama de produtos da própria Amazon, como AWS Lambda, AWS Mobile Hub e também o Amazon CloudWatch (BARE, 2016).

Para o desenvolvimento do bot dentro da plataforma existem alguns conceitos que devem ser utilizados frequentemente pelo programador. As intenções, assim como no Watson e no Dialogflow, representam os objetivos do usuário. Os enunciados são as frases que o usuário pode digitar para que uma determinada intenção seja disparada. Os *slots* representam os parâmetros que devem ser inseridos pelo utilizador para complementar uma intenção. Podem variar de acordo com cada caso de uso, mas é comum representarem objetos, cidades, datas, entre outros. Para orientar o usuário a fornecer informações para os *slots*, existem perguntas que podem ser programadas através do *prompt*. Nesse caso, o assistente solicita ao utilizador alguma informação extra para que os parâmetros de uma intenção sejam completados. Existem ainda os *fulfillments*, que são integrações que podem ser desenvolvidas pelo programador para complementar a execução de uma intenção do usuário. Representam ações externas que eventualmente são necessárias para uma aplicação deste tipo, como por exemplo agendar um horário em um calendário, ou extrair uma informação de algum banco de dados (BARE, 2016).

2.3.3.5 Wit.ai

Desenvolvida pelo Facebook, esta plataforma atualmente conta com mais de duzentos mil desenvolvedores ao redor do mundo que utilizam os seus recursos. Permite a criação e desenvolvimento de aplicações conversacionais com suporte para diversos idiomas. A plataforma aprende a cada iteração com os usuários e compartilha esse conhecimento com todos os desenvolvedores, tornando a solução cada vez mais complexa e robusta (WIT, 2020).

Atualmente suporta a interação com quatro linguagens: Ruby, Python, Node.js e Go. O algoritmo de processamento de linguagem natural opera extraíndo o significado de cada frase do usuário de maneira individual. Não existem mecanismos de identificação de contextos entre duas frases inseridas em uma mesma conversa. Portanto, caso seja de interesse do programador, os contextos devem ser tratados durante o desenvolvimento da sua própria aplicação. Além disso, não são disponibilizadas integrações com outras plataformas de maneira simplificada, cabendo ao desenvolvedor implementar esta etapa (CANONICO; DE RUSSIS, 2018).

3 PROPOSTA

A ideia de desenvolver um assistente virtual para servir como uma plataforma de interação entre os colaboradores da empresa e uma ferramenta de SIEM surgiu com base em reclamações relatadas pelos usuários. Frequentemente, membros da equipe que realizam o suporte de TI da empresa recebem dúvidas e solicitações dos colaboradores para extrair dados que estão armazenados no SIEM.

Após reuniões entre as equipes que fazem parte da infraestrutura de TI, optou-se pela criação de um assistente virtual do tipo chatbot para abordar o problema. Visto que, nesta área da empresa ainda não se havia implementado uma solução semelhante, decidiu-se que esta seria uma oportunidade de estudo e conhecimento do potencial deste tipo de aplicação. Portanto, o assistente desenvolvido neste projeto deve ser capaz de solucionar o problema de acesso à informação contida no SIEM, democratizando esta prática para que diversos colaboradores possam passar a utilizar a ferramenta de maneira autônoma. Deseja-se também analisar a eficiência e também o esforço do desenvolvimento de chatbots para outros casos internos. Dentre estes casos, destacam-se as situações onde existe uma alta demanda de informações de algum tipo de software ou programa, mas a acessibilidade destas informações é diminuída devido à necessidade do conhecimento específico daquele ambiente.

3.1 ARQUITETURA DO CHATBOT

A arquitetura proposta para o chatbot compreende a utilização de um conjunto de ferramentas para o desenvolvimento de aplicações WEB. A figura 7 representa de maneira visual alguns dos principais conceitos desta arquitetura. As escolhas das ferramentas foram feitas para que a aplicação funcione de maneira modular, sem a necessidade de restrição à somente um tipo de tecnologia ou fabricante.

O *back-end* da aplicação é o principal componente da arquitetura. Ele é responsável por tratar as mensagens do usuário, realizar o processo de autenticação, fazer pesquisas de informações no SIEM e ainda disponibilizar ao usuário respostas com dados relevantes às suas perguntas. Para realizar estas funções optou-se por utilizar a linguagem de programação Python, devido à familiaridade do autor e também da ampla gama de bibliotecas disponíveis. A biblioteca Flask foi utilizada por se tratar de um *framework* de desenvolvimento WEB, facilitando assim, o processo de criação deste tipo de aplicação. Ela foi escolhida devido à sua modularidade, simplicidade e também pela ampla documentação disponível.

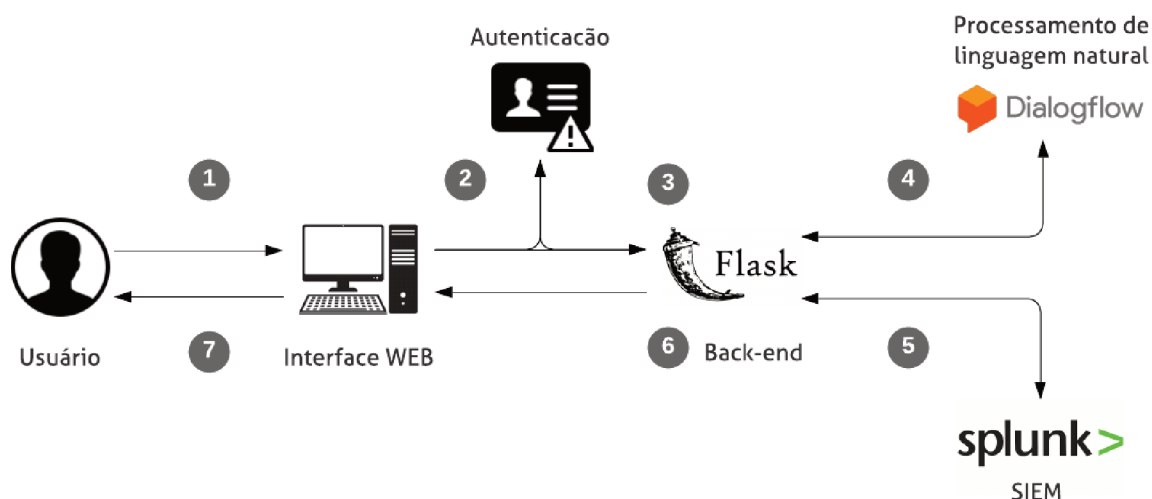
A solução de interpretação de linguagem natural selecionada foi o Dialogflow. É através desta ferramenta que são criados os diálogos e também programadas as intenções dos usuários. A escolha desta solução foi realizada em razão da simplicidade de programação e também do grande número de materiais de tutoriais disponíveis

relacionados à plataforma. Além disso, a existência de afinidades entre pessoas das duas empresas, WEG e Google, foi levada em consideração por ser um fator facilitador.

O SIEM selecionado para extrair as informações que serão repassadas aos usuários é o Splunk. Esta ferramenta já era utilizada na empresa e foi considerada a principal motivadora do projeto. Atualmente, o Splunk armazena diversas informações de várias outras aplicações que rodam na infraestrutura da empresa. Portanto, é possível implementar funcionalidades no chatbot que acessem estas informações para que em um segundo momento seja possível repassar aos usuários os dados lá contidos de maneira simplificada.

Para que o usuário possa interagir com o assistente é necessário que haja uma interface amigável e que permita a disponibilização das informações de diversas maneiras. Por este motivo, optou-se por desenvolver uma interface nova e não por utilizar outros aplicativos de chat que já permitem essa interligação. As linguagens selecionadas para a criação desta parte da arquitetura são Javascript, HTML e CSS. Através desse conjunto de linguagens torna-se possível a apresentação dos dados em diferentes formatos visuais e também cria-se liberdade para implantação de mecanismos para facilitar a utilização do assistente. Dentre estes mecanismos estão botões e cartões de sugestões de funcionalidades que servem para facilitar a utilização de algumas soluções do chatbot. Estes recursos são apresentados ao usuário em determinados momentos da utilização do assistente e podem ser configurados para serem alterados de acordo com cada caso de uso.

Figura 7 – Diagrama da arquitetura do chatbot.



Fonte – Elaborado pelo autor.

Na figura 7, pode-se verificar o funcionamento simplificado dos mecanismos da arquitetura através de uma interação com o usuário, conforme as seguintes etapas:

- (1) O usuário acessa a página WEB do chatbot.
- (2) Realiza-se o processo de autenticação.
- (3) O usuário digita uma mensagem no chat e esta é encaminhada ao *back-end* do chatbot
- (4) A mensagem é enviada pelo *back-end* ao Dialogflow para que seja possível extrair os parâmetros chave dela.
- (5) Com base na resposta recebida do Dialogflow, o *back-end* processa as informações e realiza uma pesquisa no Splunk.
- (6) Os resultados da pesquisa do Splunk são padronizados e enviados à interface WEB.
- (7) As informações são apresentadas ao usuário em formato adequado para a visualização.

3.2 ESPECIFICAÇÕES DE REQUISITOS

Os requisitos de sistema são considerados as propriedades que um software deve possuir para solucionar um ou mais problemas. Os requisitos que definem o que o sistema deve fazer são chamados de requisitos funcionais. Já as restrições e propriedades aplicadas a estes requisitos são chamadas de requisitos não funcionais (VALENTE, 2020). Para este projeto o levantamento de requisitos foi feito com base em reuniões com equipes de colaboradores para discussão das principais funcionalidades que deveriam ser implementadas. Com base nisto e no posterior refinamento das ideias, pôde-se levantar os principais requisitos da aplicação.

3.2.1 Requisitos funcionais

Os requisitos funcionais são as declarações dos serviços que o sistema deve fornecer, a forma como deve se comportar em casos específicos e como deve reagir de acordo com as diferentes entradas (SOMMERVILLE, 2011, p.59). Os requisitos funcionais levantados para esta aplicação são listados abaixo.

1. O assistente deve possibilitar a comunicação com o usuário através de mensagens no formato de texto.
2. O assistente deve orientar o usuário sobre as suas funcionalidades disponíveis.

3. O assistente deve possibilitar a extração de informações do SIEM da empresa e disponibilizá-las em formato de fácil visualização.
4. O assistente deve ser capaz de conduzir o usuário às suas funcionalidades quando não possuir resposta para uma determinada pergunta.
5. O sistema deve permitir ao usuário realizar avaliações das suas conversas com o assistente.

3.2.2 Requisitos não funcionais

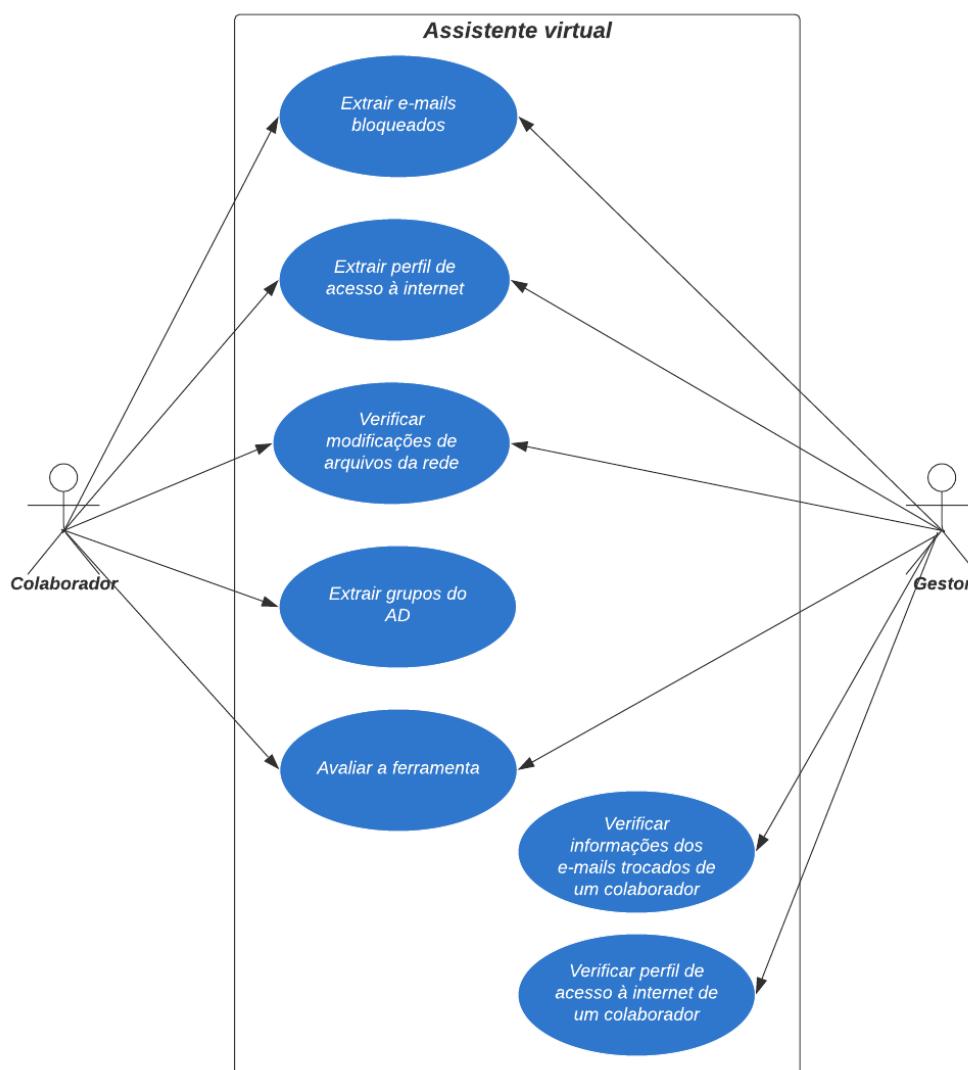
Os requisitos não funcionais correspondem a maneira como as funcionalidades devem ser executadas dentro de uma aplicação, ou seja, são as restrições de como as funções são executadas naquele sistema (WAZLAWICK, 2013). Os requisitos não funcionais deste projeto são listados abaixo.

1. Utilização da linguagem de programação Python para criação da estrutura do *back-end*.
2. O sistema deve poder ser escalável para que futuramente possam ser adicionadas novas funcionalidades.
3. Utilizar a base de dados interna de colaboradores da empresa para realizar a autenticação do usuário.
4. O sistema deve ter uma interface responsiva que possa ser visualizada em celulares, tablets, notebooks e computadores.
5. A comunicação entre o assistente e os outros softwares que compõem a sua estrutura deve ser realizada por meio de serviços WEB.
6. Utilizar o conjunto de linguagens de marcação e programação Javascript, HTML e CSS para o desenvolvimento da interface do usuário.

3.3 CASOS DE USO

Os casos de uso representam as possibilidades de interação com o sistema. Estes identificam as relações individuais entre o sistema e seus usuários ou outros sistemas. São representados através de interações e de atores que estão envolvidos com estas interações. Cada interação deve ser suplementada por informações adicionais, em formato de texto ou gráfico, com intuito de esclarecer o seu objetivo (SOMMERVILLE, 2011). Os casos de uso do assistente virtual desenvolvido neste projeto podem ser visualizados na figura 8.

Figura 8 – Diagrama de casos de uso do assistente virtual.



Fonte – Elaborado pelo autor.

3.3.1 Descrição dos casos de uso

- **Extrair e-mails bloqueados:** O usuário do tipo gestor ou colaborador solicita ao assistente a extração dos seus e-mails bloqueados pelo serviço de entrega de e-mails da empresa.
- **Extrair perfil de acesso à internet:** O usuário do tipo gestor ou colaborador solicita a verificação do seu perfil de uso de internet dentro da empresa.
- **Verificar modificações de arquivos da rede:** O usuário do tipo gestor ou colaborador solicita a verificação das pessoas que realizaram alteração em um arquivo localizado na rede compartilhada da empresa.

- **Extrair grupos do AD:** O usuário do tipo gestor ou colaborador solicita a extração dos grupos aos quais ele pertence no AD.
- **Avaliar a ferramenta:** O usuário do tipo gestor ou colaborador realiza a avaliação dos serviços do chatbot.
- **Verificar informações dos e-mails trocados de um colaborador:** O usuário do tipo gestor solicita a verificação dos e-mails trocados de um dos colaboradores da sua equipe.
- **Verificar perfil de acesso à internet de um colaborador:** O usuário do tipo gestor solicita a verificação do perfil de acesso à internet de um dos colaboradores da sua equipe.

4 FERRAMENTAS E TECNOLOGIAS

Para possibilitar o desenvolvimento deste projeto, são necessárias diversas ferramentas e tecnologias que têm como função viabilizar a criação das funcionalidades do assistente virtual. Neste capítulo, discute-se estes mecanismos através de uma descrição mais detalhada dos principais componentes do projeto.

4.1 DIALOGFLOW

O Dialogflow é uma plataforma desenvolvida pela Google que possibilita o desenvolvimento de bots através da interpretação de linguagem natural. Por meio da interface WEB é possível configurar uma série de padrões adaptáveis para diversos tipos de casos de uso, viabilizando a criação de agentes únicos que podem solucionar problemas em vários segmentos. Os agentes, como são chamados os bots dentro da plataforma, representam um módulo de alto nível que consegue processar, entender e responder mensagens de texto de um usuário.

4.1.1 Intenções

Os objetivos do usuário são representados pelas intenções. Quando um usuário deseja extrair uma informação do assistente, ele iniciará com uma pergunta. A finalidade por trás desta pergunta é exatamente o que o Dialogflow categoriza como uma “Intenção”.

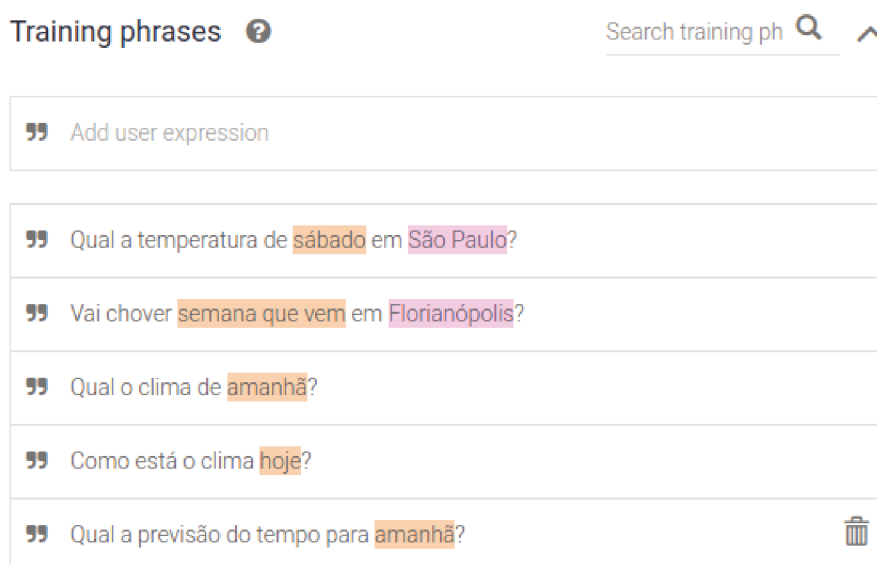
Supondo um agente que responde perguntas relacionadas à previsão do tempo. Espera-se que o usuário questione o chatbot com perguntas do tipo: “Qual a previsão do tempo para amanhã?”. Conhecendo este tipo comportamento por parte do usuário, o programador pode criar uma intenção dentro do seu agente relacionada a extração de informações meteorológicas. Desta forma, todas as perguntas referentes ao mesmo tema, neste caso clima, estarão agrupadas em um mesmo contêiner de informações chamado de “Intenção”.

4.1.1.1 Frases de treinamento

As frases de treinamento são parte essencial do mapeamento de intenções. É através delas que o agente será capaz de identificar se uma frase inserida pelo usuário se assemelha com as frases de treinamento configuradas pelo programador e com isso corresponder esta frase a uma intenção específica. Através dos algoritmos de inteligência artificial da plataforma, são criadas comparações para categorizar semelhanças entre as frases pré-configuradas e as frases inseridas. Isso faz parte do mecanismo de interpretação de linguagem natural da plataforma e é a principal característica deste tipo de solução. Desta maneira, é possível que uma frase que não

foi pré-programada possa ser mapeada para uma intenção. Este tipo de solução possibilita ao programador configurar somente algumas frases que englobam de maneira geral o objetivo do usuário e não as infinitas possibilidades de mensagens que podem ser inseridas. Um conjunto de frases de treinamento para o desenvolvimento de um assistente que responde perguntas relacionadas à previsão do tempo pode ser visto na figura 9.

Figura 9 – Exemplo de frases de treinamento para uma intenção de extração de informações do clima.



Fonte – Elaborado pelo autor.

4.1.1.2 Ação

Cada intenção pode ser configurada para disparar uma ação ao ser acionada. Esta configuração possibilita ao programador saber o momento em que uma intenção foi de fato ativada e com isso realizar algum procedimento específico para aquela situação dentro da sua aplicação.

Todas as mensagens enviadas para o Dialogflow podem ser interpretadas através de objetos JSON (do inglês, *JavaScript Object Notation*). Este tipo de notação é muito utilizada devido a facilidade de ser compreendida por humanos e também máquinas. Cada dado é representado como um conjunto de chaves e valores que podem ser interpretados por programas de qualquer tipo de linguagem.

No caso do Dialogflow, as mensagens trocadas com o assistente são demarcadas por um identificador único (*responseId*) e também por outros diversos campos que representam características extraídas da mensagem pela plataforma. Uma destas características é a ação (*action*) disparada quando uma pergunta (*queryText*) é mapeada

a uma intenção. Este comportamento pode ser visto no exemplo da figura abaixo, onde a intenção de extração de clima foi ativada e, por consequência, a ação “acao_tempo” foi disparada.

Figura 10 – Ação disparada no momento em que a intenção de extração de clima foi ativada.

```
"responseId": "db09104b-df39-455c-8ffb-6a5c1ca1a985-0f0e27e1",  
"queryResult": {  
  "queryText": "Qual a temperatura de sábado em São Paulo?",  
  "action": "acao tempo",
```

Fonte – Elaborado pelo autor.

4.1.1.3 Contexto

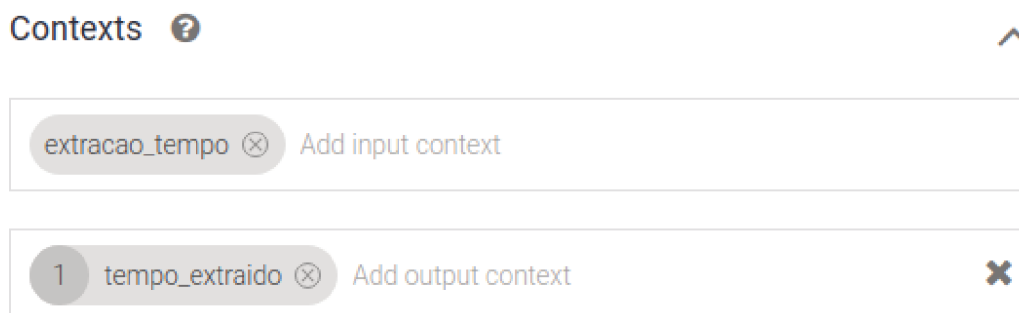
Os contextos possibilitam ao agente tornar a conversação mais complexa e, conseqüentemente, mais parecida com uma interação humana. Estes mecanismos servem para dar continuidade a um assunto específico que está sendo tratado num dado momento de uma conversa para que depois seja possível responder novas perguntas relacionadas a um tema que estava sendo discutido anteriormente.

No caso de uma pergunta relacionada ao clima da cidade de Florianópolis, um usuário pode questionar qual a previsão do tempo para sábado. O agente gera uma resposta e em seguida questiona o usuário se ele gostaria de verificar o clima para mais algum dia da semana. O usuário então solicita a previsão do clima para domingo, porém desta vez sem inserir qual a cidade. O agente, através do contexto, é capaz de identificar que o usuário está solicitando uma nova extração para a cidade de Florianópolis e por este motivo responde com as informações da previsão para domingo.

O menu de contexto está disponível em todas as intenções que são criadas no Dialogflow. Cada intenção pode ter um contexto de entrada e/ou um de saída. Os contextos de entrada servem como pré-requisito para que uma intenção seja ativada e os contextos de saída indicam que a intenção já foi ativada. Desta forma, é possível relacionar uma intenção com a outra simplesmente ligando os contextos delas para formar padrões de conversação.

Na figura 11, pode-se verificar os contextos de entrada e saída de uma intenção exemplo. Devido a este mecanismo, para ativar a intenção em questão, é necessário que uma outra intenção com contexto “extracao_tempo” seja ativada anteriormente. Caso esta condição seja satisfeita, a intenção é disparada. Além disso, após ser disparada, ela terá como saída o contexto “tempo_extraido”, que por sua vez, poderá ser utilizado por uma nova intenção para dar continuidade a conversa.

Figura 11 – Exemplo de contextos de uma intenção.



Fonte – Elaborado pelo autor.

4.1.1.4 Respostas

O campo de respostas pode ser configurado dentro de cada intenção criada. As respostas representam as mensagens de texto que o agente retorna ao usuário quando uma intenção é acionada. Estas respostas podem ter caráter de finalização, quando a mensagem encerra uma extração naquele ponto, ou de continuidade, quando o agente questiona o usuário para que novas informações sejam inseridas. O tipo de resposta depende de cada aplicação e cada caso de uso, cabendo assim ao programador avaliar se é necessário solicitar novos dados ou não.

Uma mesma intenção pode conter várias respostas que, em geral, significam a mesma informação. Isso acontece para que o agente não seja tão previsível ao responder ao usuário, alterando a resposta de tempos em tempos. A seleção de qual das respostas é escolhida dentro da lista pré-programada fica por parte do Dialogflow, que utiliza um algoritmo randômico de escolha. É possível visualizar na figura 12, a existência de três respostas que possuem a mesma finalidade, porém são descritas de maneiras diferentes. Assim, o agente varia o formato dos resultados apresentados.

Figura 12 – Exemplos de respostas de uma intenção.

Text Response	
1	A previsão é de um dia de sol
2	Será um dia ensolarado
3	Um dia com céu limpo e sem nuvens

Fonte – Elaborado pelo autor.

4.1.1.5 Intenções generalizadas

As intenções generalizadas são ativadas em momentos em que uma mensagem inserida pelo usuário não é mapeada para nenhuma intenção específica criada pelo programador anteriormente. Estas intenções servem para que o agente possa responder algo quando for questionado por meio de alguma frase que ele ainda não conhece. Este mecanismo possibilita ao desenvolvedor conduzir o usuário para as perguntas que ele pré-programou no assistente e, com isso, ter uma resposta adequada para aquela informação.

A principal intenção generalizada é conhecida como *fallback* e ela é ativada quando nenhuma das outras intenções foram acionadas. Uma forma de utilizar esta intenção é representada na figura 13, onde o programador conduz o usuário a extrair informações que estão programadas no agente, fazendo com que a conversa possa ter continuidade.

Figura 13 – Exemplo de respostas de uma intenção *fallback*.

Text Response	
1	Desculpe, no momento eu posso te ajudar a extrair informações relacionadas ao clima das cidades
2	Lamento, porém até o momento eu somente consigo fornecer informações da previsão do tempo

Fonte – Elaborado pelo autor.

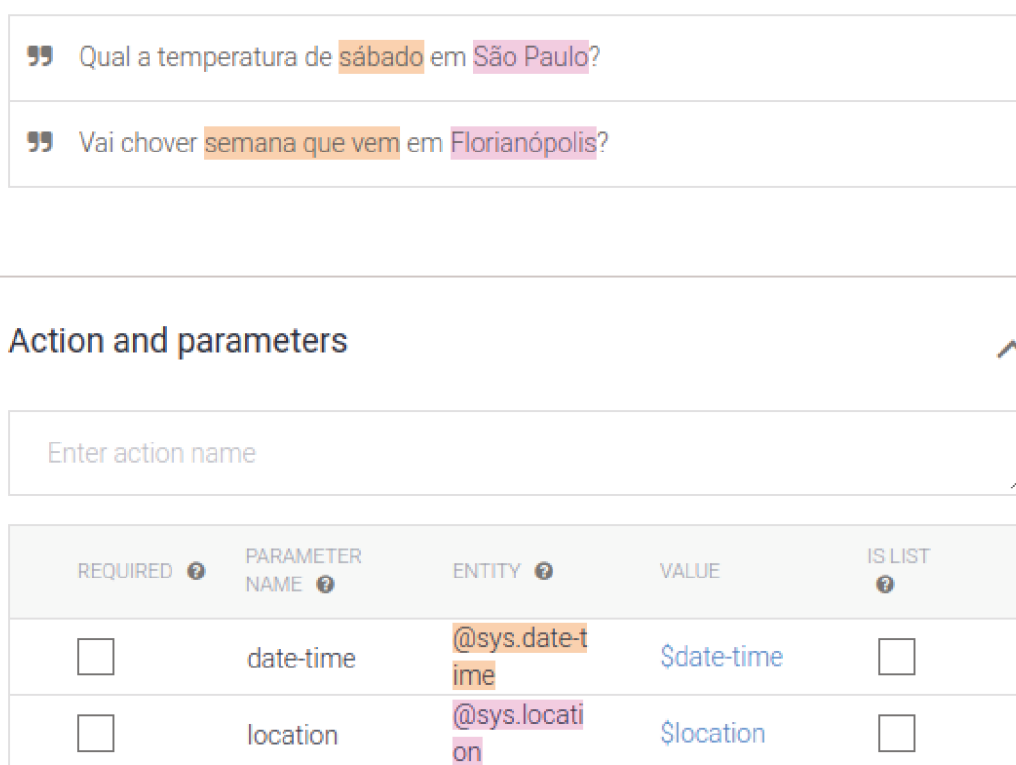
4.1.2 Entidades

Quando uma intenção é mapeada de acordo com a mensagem inserida pelo usuário, o Dialogflow pode extrair diversos parâmetros daquela intenção. Os parâmetros são úteis para possibilitar a customização de respostas mais complexas do agente para o usuário. Cada parâmetro possui um tipo, que é conhecido na plataforma como entidade. As entidades determinam qual o formato de extração de um parâmetro. Existem diversas entidades que são pré-configuradas dentro da plataforma. Dentre elas estão entidades de cidades, endereços de e-mail, datas, nomes, entre outros. Desta forma, é possível realizar extrações de parâmetros das frases do usuário e mapeá-los a uma destas entidades para personalizar a resposta ao usuário.

Em uma mensagem de previsão do tempo, é possível notar alguns parâmetros que são considerados como chaves para que a resposta seja adequada. Dentre estes parâmetros destacam-se a data e a cidade que o usuário deseja saber a previsão. Ambos os parâmetros já possuem entidades pré-configuradas e por este motivo podem

ser extraídos da maneira correta. Os parâmetros de extração são sinalizados pelo programador através da seleção da palavra chave na frase de treinamento daquela intenção e, em seguida, da escolha do tipo de entidade daquele parâmetro. Na figura 14, pode-se notar que a data da frase treinamento foi selecionada e posteriormente mapeada para a entidade “*sys.date-time*”, que corresponde a entidade de datas. O mesmo acontece para a cidade, que por sua vez, foi mapeada para a entidade de localidade “*sys.location*”.

Figura 14 – Seleção de entidades em frases de treinamento.



The screenshot shows a training interface with two example sentences and a configuration table. The first sentence is "Qual a temperatura de sábado em São Paulo?" with "sábado" highlighted in orange and "São Paulo" in purple. The second sentence is "Vai chover semana que vem em Florianópolis?" with "semana que vem" in orange and "Florianópolis" in purple. Below the sentences is a section titled "Action and parameters" with a search box and a table.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	date-time	@sys.date-time	\$date-time	<input type="checkbox"/>
<input type="checkbox"/>	location	@sys.location	\$location	<input type="checkbox"/>

Fonte – Elaborado pelo autor.

Após a configuração de parâmetros ter sido realizada nas frases de treinamento, é possível extrair estes dados por meio do objeto JSON. Desta maneira, pode-se personalizar uma resposta ao usuário com os parâmetros identificados e tornar a experiência mais complexa. Na figura 15, nota-se que as entidades de cidade e data, que haviam sido programadas anteriormente, são extraídas através de parâmetros no próprio objeto JSON. Além disso, uma ação foi disparada indicando que a intenção de extração de informações meteorológicas foi ativada.

Figura 15 – Extração de parâmetros de uma frase de treinamento por meio do objeto JSON.

```
"responseId": "0f79cddf-ac13-4de1-b26c-3e4c4cb4193f-0f0e27e1",
"queryResult": {
  "queryText": "Qual a temperatura de sábado em São Paulo?",
  "action": "acao tempo",
  "parameters": {
    "location": {
      "subadmin-area": "",
      "island": "",
      "country": "",
      "shortcut": "",
      "city": "São Paulo",
      "admin-area": "",
      "zip-code": "",
      "business-name": "",
      "street-address": ""
    },
    "date-time": "2020-05-23T12:00:00-03:00"
  }
}
```

Fonte – Elaborado pelo autor.

4.2 SPLUNK

O Splunk é uma ferramenta de SIEM (do inglês, *Security Information and Event Management*) que tem como característica gerenciar eventos de máquinas. A sua principal funcionalidade é permitir a extração de informações dos eventos indexados para gerar monitoramentos e examinar os dados para tomadas de decisão.

A ferramenta é capaz de realizar a análise de quaisquer eventos (*logs*) dos mais diversos tipos de aplicações, como sistemas operacionais, bancos de dados e softwares. Através da interface WEB, é possível realizar pesquisas para extrair dados relevantes dos *logs* da aplicação que está sendo analisada possibilitando a criação de uma rede de monitoramentos específica para cada caso. Além disso, o Splunk possibilita a geração de alertas através de pesquisas pré-programadas pelos desenvolvedores que são executadas periodicamente. Em casos de rompimento de um limiar pré-estabelecido, os alertas avisam as equipes responsáveis, através de e-mails, para que medidas sejam tomadas visando solucionar o problema.

Os painéis (*dashboards*) também fazem parte das funcionalidades da ferramenta. Através deles, é possível criar pesquisas que geram resultados em formatos gráficos ou de tabelas para facilitar a visualização dos dados para usuários que não são especialistas na plataforma. Os painéis podem ser customizados, caso o programador permita, pelo próprio usuário final, fazendo com que alguns parâmetros das pesquisas sejam alterados através de menus de fácil acesso.

Para a criação dos monitoramentos, alertas e painéis, o Splunk utiliza uma linguagem de programação própria conhecida como *Search Processing Language* (SPL).

Esta é uma linguagem similar à famosa SQL (do inglês, *Structured Query Language*) utilizada para a consulta de informações em banco de dados relacionais. Apesar de não se tratar de um banco de dados relacional propriamente dito, o Splunk é intitulado como um banco distribuído, não relacional, semi-estruturado e com uma dimensão de tempo implícita (CARASSO, 2010). Por este motivo, algumas semelhanças entre as linguagens podem ser encontradas, principalmente no modo de pensar do programador ao iniciar uma nova expressão de extração de dados.

A figura 16 demonstra a utilização da SPL para a criação de uma pesquisa dentro da plataforma do Splunk. É possível notar a complexidade dos comandos para um usuário que não possua um conhecimento específico da linguagem do software. Apesar da lógica de realização de comandos ser semelhante ao SQL, a linguagem do Splunk possui diversos comando próprios que são exclusivos desta plataforma.

Figura 16 – Exemplo de uma pesquisa realizada no Splunk através da linguagem SPL

```
1 sourcetype=WinEventLog:Security earliest =-72h@h latest=@h (EventCode=4625 OR EventCode=529 OR EventCode=530 OR
   EventCode=531 OR EventCode=532 OR EventCode=533 OR EventCode=534 OR EventCode=535 OR EventCode=536 OR
   EventCode=537 OR EventCode=539)
2 | eval NomeConta=mvindex(Account_Name,1)
3 | bin _time span=1h
4 | eventstats count as Contagem by _time
5 | eventstats avg(eval(if((_time < relative_time(now(), "-1h@h"))AND (Contagem<35),Contagem,null))) as Media std
   (eval(if((_time <relative_time(now(), "-1h@h"))AND (Contagem<35),Contagem,null))) as "DesvioPadrao"
6 | eval limitesuperior=Media+3*DesvioPadrao, limiteinferior=Media-3*DesvioPadrao
7 | where _time=relative_time(now(), "-1h@h")
8 | where Contagem > limitesuperior
9 | stats count by _time NomeConta Workstation_Name Source_Network_Address
```

Fonte – Elaborado pelo autor.

4.3 PYTHON

Python é uma linguagem de programação de alto nível com uma filosofia de simplificação que se destaca pela sua facilidade de aprendizado e potencial. Dentre os seus principais pilares estão a explicitude, a simplicidade e a legibilidade do código, tornando-a uma das mais utilizadas linguagens de programação dos últimos anos (PETERS, 2004). Devido a estes fatores é considerada como uma das principais opções para desenvolvimento de diversos tipos de aplicações, se destacando principalmente nas áreas de aprendizado de máquinas, ciência de dados e desenvolvimento WEB.

A comunidade de programadores que utiliza o Python é também uma das suas maiores forças. Devido ao grande número de usuários, possui uma imensa base de bibliotecas, criadas pelos próprios utilizadores, que facilitam o processo de desenvolvimento de novas aplicações. Pelo caráter de código aberto (*open source*), a comuni-

dade cria e contribui para projetos generalistas das mais diversas áreas que podem ser utilizados como base por outros usuários desenvolverem as suas próprias soluções específicas.

4.3.1 Flask

Os *frameworks* são códigos desenvolvidos de maneira abstrata que visam facilitar o desenvolvimento de uma funcionalidade genérica para que novos programadores possam criar os seus softwares de maneira mais simplificada e eficiente. Através da utilização deles, o processo de criação de uma nova solução é focado na configuração das funcionalidades implementadas, facilitando o desenvolvimento e reduzindo o retrabalho.

O Flask é um *framework* desenvolvido em Python para a criação de aplicações WEB. É caracterizado por apresentar um estrutura inicial simplificada, suportando apenas o necessário para a criação de uma aplicação. Entretanto, através dos seus módulos, o programador pode optar por deixar a solução cada vez mais completa e complexa, adaptando-a ao seu próprio caso de uso.

Através do Flask é possível desenvolver o *back-end* de aplicações WEB, responsável por tratar as regras de negócio do software. Dentre as suas funcionalidades pode-se destacar a formatação e validação de dados, manejo de banco de dados, extrações de dados de aplicações externas e também tratamento de rotas WEB.

4.4 JAVASCRIPT, HTML E CSS

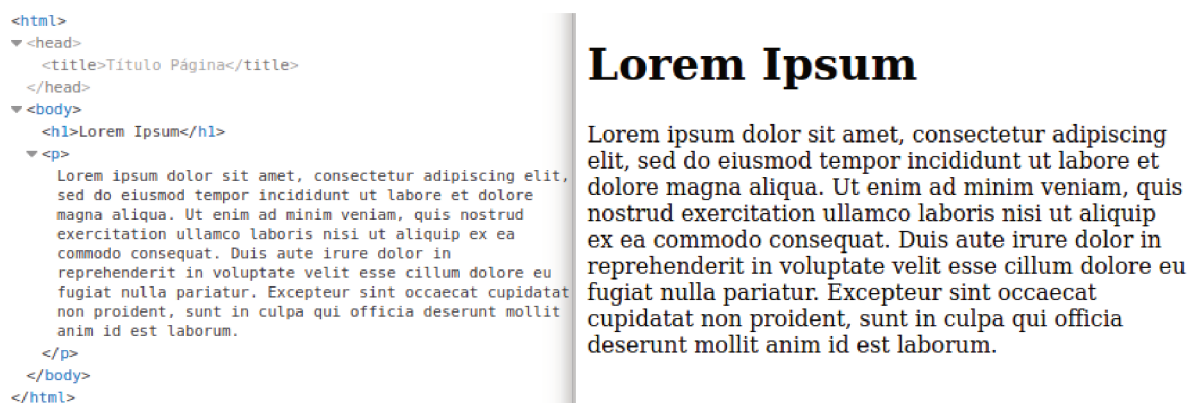
A base de programação *front-end* de aplicações WEB é composta em grande parte dos casos pelo conjunto Javascript, HTML e CSS. Estas três linguagens normalmente são utilizadas em conjunto por fornecerem uma gama de possibilidades que se complementam e tornam as interfaces de sites muito mais completas e iterativas.

O HTML se caracteriza por compor a estrutura de uma página WEB. É considerado como um linguagem de marcação, onde a formatação dos elementos é alterada de acordo com marcações pré-desenvolvidas. Ao contrário de uma linguagem de programação, o HTML não possibilita a criação de comportamentos computacionais interativos ou dinâmicos (HOPE, 2019).

Através dele, é possível posicionar elementos como textos, formulários, imagens e tabelas para que o aspecto visual da interface fique corretamente formatado de acordo com as especificações do programador. O funcionamento é dado a partir da criação de *tags* que servem para especificar os tipos de conteúdos da página WEB. É possível especificar regras para diferentes tipos de componentes, como parágrafos e cabeçalhos, moldando assim a estrutura de diferentes maneiras. A figura 17 representa uma estrutura simplificada de um código HTML, onde são especificados um cabeçalho

e um parágrafo de uma página. No lado direito da figura, é possível visualizar o resultado do código desenvolvido quando aberto em um navegador WEB. Pode-se notar claramente a separação dos elementos de acordo com os padrões especificados no código.

Figura 17 – Exemplo de código HTML para criação de uma página WEB.



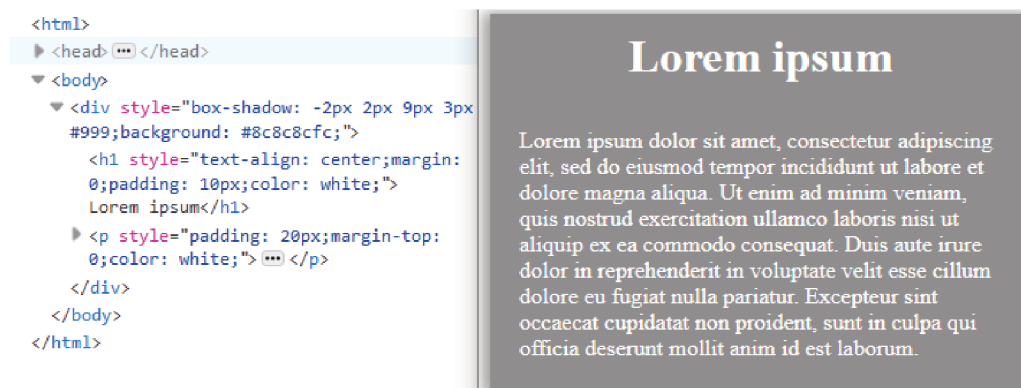
Fonte – Elaborado pelo autor.

O código HTML funciona muito bem para estruturar a formatação da página, entretanto para deixá-la mais agradável para o usuário, torna-se necessário utilizar o CSS. Esta linguagem permite estilizar os componentes HTML e, com isso, personalizar estruturas especificadas fazendo com que o programador tenha mais liberdade para criar a página com diversos componentes únicos.

O CSS se baseia num conjunto de regras que são associadas aos elementos HTML da página. Estas regras controlam como os elementos serão renderizados quando o usuário abrir a página em seu navegador. Cada regra possui um seletor, que representa um elemento HTML, e também um par de propriedade e valor, que por sua vez, irão alterar uma característica específica daquele elemento escolhido pelo seletor (DUCKETT, 2011). No lado esquerdo da figura 18 são aplicados alguns elementos CSS no mesmo código mostrado na figura 17. Já no lado direito, pode-se notar o resultado através da formatação centralizada do título, alteração da cor de fundo e também a aplicação de sombra nas bordas.

Com a utilização em conjunto dos elementos HTML e CSS é possível criar páginas WEB para atender a grande maioria dos requisitos visuais de uma interface para o usuário. Entretanto, estas linguagens trabalham por característica com a parte estática da página, ou seja, como os elementos são apresentados ao usuário. Para adicionar funcionalidades dinâmicas à interface é necessário a utilização de uma outra linguagem que possa interagir com os elementos já criados. Uma das possíveis soluções é a utilização do Javascript, que por ser uma linguagem de programação, permite ao

Figura 18 – Exemplo de código CSS para estilização de uma página WEB.



Fonte – Elaborado pelo autor.

desenvolvedor implementar dinamismo à página WEB, deixando a solução muito mais complexa.

Através do Javascript torna-se possível realizar ações como inserir novos elementos na página dinamicamente, manipular dados, reagir a eventos, determinar o tamanho da tela do usuário e também performar ações com base em condições (DUC-KETT, 2011). Todas essas características auxiliam o programador a ter mais liberdade para criar soluções mais funcionais.

Uma das principais características do Javascript neste formato de implementação, é fato de que todo o código desenvolvido passa a ser executado no navegador do usuário. Desta forma, todo o processamento que é feito na interface, é gerenciado pelo próprio computador do usuário, através do seu navegador. Este tipo de solução permite que o servidor no qual o *back-end* da aplicação roda, fique livre para realizar outros processamentos, diminuindo assim eventuais sobrecargas.

5 DESENVOLVIMENTO

Durante a implementação das funcionalidades do assistente virtual é necessário desenvolver diversos mecanismos para solucionar os problemas que surgem no decorrer da execução do projeto. Neste capítulo, trata-se da discussão e das abordagens de desenvolvimento dos recursos do chatbot. É feita também uma análise de funcionamento da arquitetura completa para um caso de uso.

5.1 INTERFACE DO USUÁRIO

A interface de usuário, também conhecida como *front-end* da aplicação, foi criada com base em uma interface do tipo chat. A ideia principal é que o usuário possa inserir as suas mensagens em uma caixa de texto e em seguida enviá-las através do seu teclado ou também de um botão na tela. As mensagens enviadas e recebidas devem ser mostradas de maneira clara por meio de um painel. Neste painel deve-se ficar destacado evidentemente se a mensagem é de autoria do usuário ou se foi uma resposta do chatbot. Além disso, outro requisito levantado está na possibilidade de as mensagens do chatbot poder serem apresentadas não somente no formato de texto, mas também através de gráficos e tabelas para facilitar a visualização dos dados extraídos. Quando o assistente realiza a etapa de processamento das mensagens e também de busca de informações para gerar respostas, deve-se deixar claro ao usuário que o chatbot está ativo.

Com base nos requisitos apresentados, foi desenvolvida uma interface através da combinação das linguagens HTML, CSS e Javascript. Toda a parte visual e estática é criada através do processamento de recursos HTML que moldam a estrutura da página do chatbot. Botões, caixas de texto e o painel de visualização de mensagens são exemplos de recursos HTML criados para moldar a interface que o usuário visualiza ao entrar no assistente. A estilização de todos os componentes da interface é realizada através de especificações com CSS que permitem personalizar os elementos através da aplicação de cores, sombras e modificação de tamanhos dos componentes para tornar a experiência do usuário muito mais agradável.

Para tornar a interface responsiva às ações do usuário foi desenvolvido um programa em Javascript que é responsável por tratar interações. Este programa permite que atividades como clicar em um botão, redimensionar a tela do navegador e pressionar uma tecla do teclado gerem respostas na interface. Essas interações deixam a experiência de utilização mais completa, facilitando o processo de utilização do assistente.

Ao entrar na página WEB do chatbot pela primeira vez, o usuário é orientado a pressionar um dos botões dos cartões de sugestões de funcionalidades. Estes cartões foram desenvolvidos para que o usuário saiba, sem nenhum pré-requisito, quais são

as principais funcionalidades do assistente. Dessa maneira, ele pode selecionar um dos cartões já sabendo o que esperar do resultado final daquela interação. Ao longo das conversas com o chatbot, são apresentados ao usuário botões de sugestão que aparecem logo acima da caixa de texto onde o utilizador pode digitar a sua mensagem. Estes botões possuem funcionalidades diferentes de acordo com a etapa da conversa e podem apresentar informações como calendários para escolha de datas ou também pequenos vídeos com tutoriais para extrair dados. Na figura 19, nota-se a primeira mensagem do assistente (1), orientando o usuário a realizar a escolha de uma das opções dos cartões de sugestão. Logo abaixo da mensagem, são apresentados os próprios cartões com informações da principais funcionalidades do assistente (2). Em seguida, o usuário pressiona um dos botões dos cartões de sugestão e a mensagem relativa àquele cartão é apresentada na tela (3). O assistente responde solicitando a inserção de uma data inicial de extração dos dados. Somente neste momento o botão de sugestão “Abril calendário” (4) é mostrado ao usuário, que pode optar por clicar nele ou não. Caso o usuário opte por clicar no botão, um calendário é mostrado e a data pode ser selecionada. Caso o usuário prefira digitar a data manualmente, ele pode realizar este processo através da caixa de texto (5). Ao longo das diversas possibilidades de interação entre o usuário e o assistente, estes recursos de sugestão são apresentados na interface visando facilitar o processo de utilização da ferramenta.

Figura 19 – Interface de usuário do assistente virtual

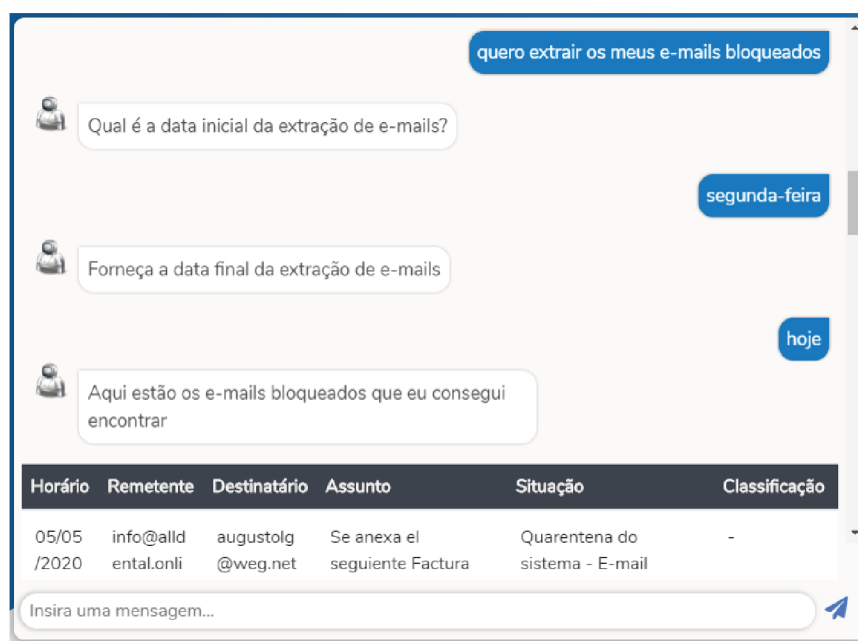


Fonte – Elaborado pelo autor.

5.2 PRINCÍPIO DE FUNCIONAMENTO

Para que seja possível realizar a extração de informações do Splunk e fornecer ao usuário estes dados de maneira simplificada, é necessário realizar um processamento das informações que são inseridas no assistente a cada iteração. Neste tipo de aplicação, cada iteração pode ser considerada uma mensagem que o usuário digita ou até algum botão pressionado. É preciso que haja uma validação da mensagem para que a intenção do usuário possa ser transformada em uma consulta na plataforma de SIEM, resultando em um conjunto de informações apresentadas em um formato adequado de interpretação. Uma iteração completa com o assistente pode ser visualizada na figura 20, onde o usuário realiza uma consulta dos e-mails que não foram entregues à sua caixa de entrada.

Figura 20 – Interação com o assistente para extração de e-mails bloqueados.



Fonte – Elaborado pelo autor.

Ao entrar na página do chatbot, o usuário é orientado a iniciar a conversa através dos cartões de sugestão. Estes cartões servem principalmente para informar quais são as funcionalidades do assistente e com isso melhorar experiência de utilização. Lendo as informações dos cartões, o usuário consegue entender rapidamente o que ele pode questionar ao assistente e assim iniciar a extração de uma informação.

Ao clicar no botão de um dos cartões de sugestão, ou digitar algum termo relacionado ao conteúdo dos cartões, o usuário começa o processo de interação com o assistente. Cada mensagem inserida através do chat é repassada ao servidor, onde são feitas as análises. Ao chegar no *back-end* da aplicação, a mensagem é enviada

para o Dialogflow para que seja possível realizar o processamento em linguagem natural e, por consequência, a extração da intenção do usuário. Este processo é realizado através da *Application Programming Interface* (API) do Dialogflow. Este tipo de solução serve como uma ferramenta para troca de informações entre softwares, neste caso a plataforma do Google e o chatbot. Por meio da API, a mensagem do usuário é enviada para o Dialogflow e a plataforma retorna uma resposta através de um objeto do tipo JSON com informações relacionadas ao processamento daquela mensagem, conforme a figura 21. Todos os dados e intenções extraídos neste momento são consequência da pré-programação que foi realizada dentro do Dialogflow por meio da interface WEB. Desta maneira, ao receber uma mensagem, a plataforma é capaz de mapeá-la à uma intenção criada anteriormente pelo programador. Como é possível notar na figura 21, a mensagem do usuário (*query_text*), a intenção mapeada (*action*) e também a resposta do assistente (*fulfillmentText*) ficam disponíveis através do objeto JSON enviado como resposta à solicitação feita à API do Dialogflow. Assim, é possível utilizar estas informações no *back-end* do chatbot para poder encaminhar ao usuário a resposta indicada e também identificar a intenção mapeada.

Figura 21 – Resposta JSON do Dialogflow à primeira iteração do usuário.

```
"responseId": "1d3049f3-1cb1-44df-bb3a-fd3e99aa4d15-0f0e27e1",
"queryResult": {
  "queryText": "quero extrair os meus emails bloqueados",
  "action": "extracao-de-emails-bloqueados",
  "parameters": {
    "date-time": "",
    "date": "",
    "date1": ""
  },
  "fulfillmentText": "Por favor, forneça a data inicial da extração de e-mails",
  "fulfillmentMessages": [
    {
      "text": {
        "text": [
          "Por favor, forneça a data inicial da extração de e-mails"
        ]
      }
    }
  ]
},
```

Fonte – Elaborado pelo autor.

O usuário recebe então uma resposta do assistente relacionada à sua solicitação inicial. Neste caso, o assistente sugere que seja inserida uma data inicial para a extração das informações de e-mail. Assim que a resposta é inserida, realiza-se novamente o envio da mensagem do usuário à API do Dialogflow para que seja possível extrair as informações necessárias até que todos os parâmetros obrigatórios

sejam satisfeitos. Os parâmetros obrigatórios são sinalizados na configuração do Dialogflow quando uma intenção é criada. Eles são considerados essenciais para que posteriormente seja possível converter as mensagens do usuário em uma extração do Splunk. Na figura 22, pode-se verificar que o usuário insere a data inicial (*query_text*) e o Dialogflow mapeia esta mensagem ao primeiro parâmetro obrigatório (*date*) desta intenção. Neste caso, a ferramenta foi capaz de interpretar a palavra “segunda-feira” como sendo uma data da semana atual e realizar a conversão para o formato apropriado. Ainda na mesma figura é possível notar a resposta do assistente após essa iteração através do campo “*fulfillmentText*”.

Figura 22 – Resposta JSON do Dialogflow à segunda iteração do usuário.

```
"responseId": "66cd6384-586c-444f-8c83-8910a7b0b8ab-0f0e27e1",
"queryResult": {
  "queryText": "segunda-feira",
  "action": "extracao-de-emails-bloqueados",
  "parameters": {
    "date": "2020-06-01T12:00:00-03:00",
    "date1": "",
    "date-time": ""
  },
  "fulfillmentText": "Por favor, forneça a data final da extração de e-mails",
  "fulfillmentMessages": [
    {
      "text": {
        "text": [
          "Por favor, forneça a data final da extração de e-mails"
        ]
      }
    }
  ]
},
```

Fonte – Elaborado pelo autor.

A mensagem de resposta do assistente é enviada ao usuário que, por sua vez, pode inserir a data final da extração. Quando o usuário responde ao assistente, a mensagem (“*queryText*”) é novamente processada pela API do Dialogflow. Ela é mapeada ao parâmetro “*date1*”, conforme a figura 23. Nota-se que todos os parâmetros obrigatórios foram satisfeitos por meio do campo “*allRequiredParamsPresent*”.

Quando todos os parâmetros obrigatórios são inseridos realiza-se, no *backend*, um mapeamento das informações recebidas do Dialogflow. Estes dados são convertidos em uma expressão de consulta na linguagem SPL da plataforma Splunk. Através desta conversão, juntamente com os parâmetros do Dialogflow, pode-se efetuar a consulta no Splunk, por meio de uma API do software. A resposta recebida pela API corresponde aos dados que devem ser apresentados ao usuário. Os dados são convertidos para um formato adequado de visualização como tabelas, gráficos ou

mesmo uma mensagem de texto. Após este processo as informações são enviadas ao usuário.

Figura 23 – Resposta JSON do Dialogflow à terceira iteração do usuário

```
"responseId": "d8d6a585-1e0a-4e62-aeee-e0825bc233fb-0f0e27e1",
"queryResult": {
  "queryText": "hoje",
  "action": "extracao-de-emails-bloqueados",
  "parameters": {
    "date1": "2020-05-26T12:00:00-03:00",
    "date": "2020-06-01T12:00:00-03:00",
    "date-time": ""
  },
  "allRequiredParamsPresent": true,
  "fulfillmentText": "Escolha uma das opções abaixo se você quiser extrair uma nova
informação",
  "fulfillmentMessages": [
    {
      "text": {
        "text": [
          "Aqui estão os e-mails bloqueados que eu encontrei"
        ]
      }
    }
  ],
}
```

Fonte – Elaborado pelo autor.

5.3 FUNCIONALIDADES

As funcionalidades do chatbot são escolhidas com base em um levantamento feito em conjunto com os colaboradores que realizam a função suporte de TI na empresa. Estes colaboradores são considerados como elementos chaves para esta discussão, devido ao contato frequente com os usuários para solucionar os mais diversos problemas de TI.

Somente as principais demandas dos usuários foram selecionadas para se tornarem funcionalidades do chatbot. Esta escolha foi realizada devido ao trabalho de implementação de cada opção, já que demandas não tão recorrentes não obteriam tanto engajamento por parte dos usuários e, portanto, poderiam continuar sendo resolvidas pela equipe de suporte.

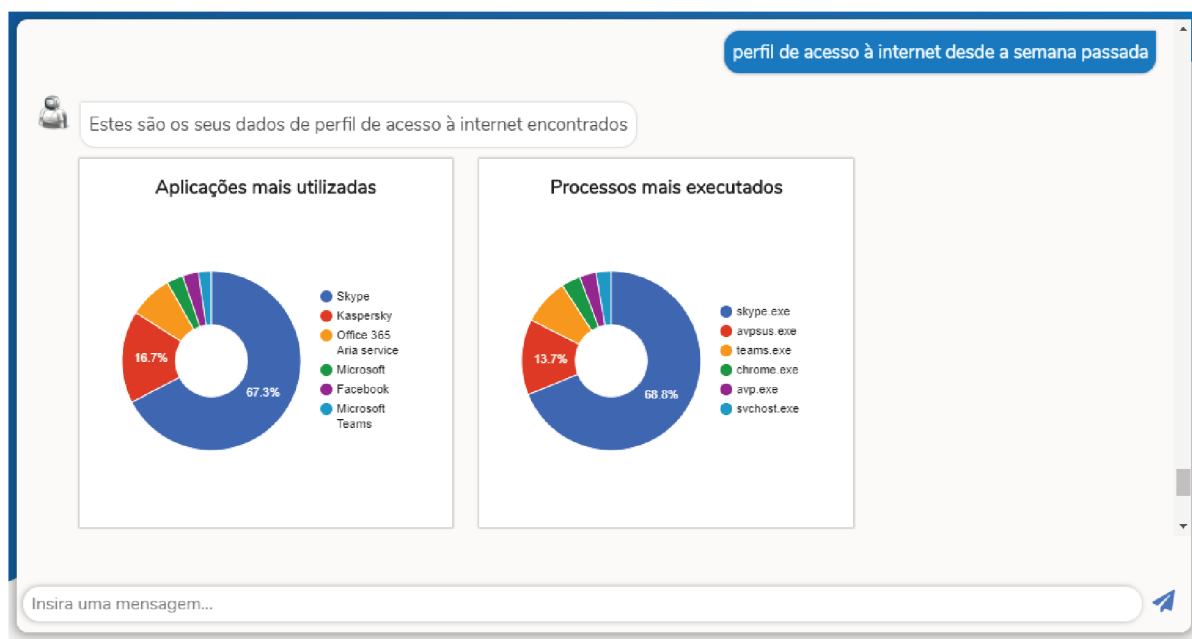
5.3.1 Extração de perfil de acesso à internet

O acesso à internet da empresa é gerenciado por um dispositivo que realiza verificações de ameaças e controles para garantir a segurança dos usuários durante a navegação. Este dispositivo permite realizar o armazenamento de eventos de uso de

internet de qualquer usuário, gerando assim uma base de dados com informações relativas ao perfil de acesso à internet de cada colaborador. Com base nestes eventos que estão armazenados no Splunk, torna-se possível realizar uma análise dos tipos de aplicação, como cliente de e-mail, navegador de internet e aplicativos de teleconferência que mais consumiram dados de internet para cada usuário.

Para iniciar a extração deste tipo de informação através do chatbot é necessário que o usuário realize uma solicitação de consulta de perfil de acesso à internet. Em seguida o assistente solicita uma data inicial e final para que a consulta seja realizada. Assim que o usuário insere os dados solicitados, o assistente busca as informações relacionadas ao perfil daquele usuário em específico e, então, gera dois gráficos únicos. O primeiro gráfico representa quais foram os principais aplicativos do computador daquele usuário que mais consumiram dados de internet. Já o segundo gráfico demonstra quais são os processos executados que mais utilizaram dados de internet. Um modelo de exemplo de conversa para extração do perfil de acesso à internet pode ser visto na figura 24.

Figura 24 – Conversa exemplo para extração do perfil de acesso à internet.



Fonte – Elaborado pelo autor.

5.3.2 E-mails bloqueados

A troca de e-mails é uma das principais opções de comunicação entre os colaboradores da empresa e também com usuários externos, como fornecedores e clientes. Todos os e-mails que são trocados externamente, ou seja, fora da rede interna da

empresa, são analisados por uma ferramenta de anti-spam. Este tipo de ferramenta realiza uma análise automatizada em diversas características de um e-mail, como remetente, destinatário, corpo de texto, anexo, entre outros. Com base nesta análise a ferramenta é capaz de identificar um e-mail como sendo potencialmente perigoso para o usuário e, portanto, coloca ele em uma caixa de quarentena ou até mesmo não realiza a entrega dele.

Devido a este tipo de política de bloqueio de e-mails como caráter perigoso, os usuários frequentemente não recebem alguns e-mails trocados com outras empresas e pessoas externas. Por este motivo, os usuários acabam solicitando ao suporte de TI da empresa uma análise de quais e-mails trocados por eles foram colocados na quarentena ou até bloqueados.

Com base nesta demanda, optou-se por implementar esta funcionalidade no assistente. Quando o usuário realiza a solicitação para extrair os seus e-mails bloqueados, o assistente pede para que seja inserido o período de realização da verificação. Assim que o usuário insere os dados requisitados, o assistente retorna, em formato de tabela, os e-mails que foram enviados para o usuário, mas que não foram entregues. Na própria tabela é possível verificar quem foi o remetente do e-mail, a data de recebimento e também dois campos indicando quais os motivos do e-mail não ter sido entregue. A conversa exemplo pode ser vista na figura abaixo.

Figura 25 – Conversa exemplo para extração dos e-mails bloqueados.

Horário	Remetente	Destinatário	Assunto	Situação	Classificação
05/05/2020 15:02	info@alldental.online	augustolg@weg.net	Se anexa el siguiente Factura Electrónica :ATT 1368481880 Data: 05/05/2020	Quarentena do sistema - E-mail recuperável mediante contato com suporte	-
31/03/2020 16:45	esilva@techsupply.com.br	augustolg@weg.net	RE: Versionamento Scripts Criados no ACL	Quarentena do usuário - E-mail recuperável mediante desbloqueio na caixa de entrada	-
29/03/2020 11:26	update@go.facebook.com	augustolg@weg.net	Atualizações da API do WhatsApp Business	Bloqueado - E-mail não pode ser recuperado	Remetente do e-mail está bloqueado pelo sistema

Fonte – Elaborado pelo autor.

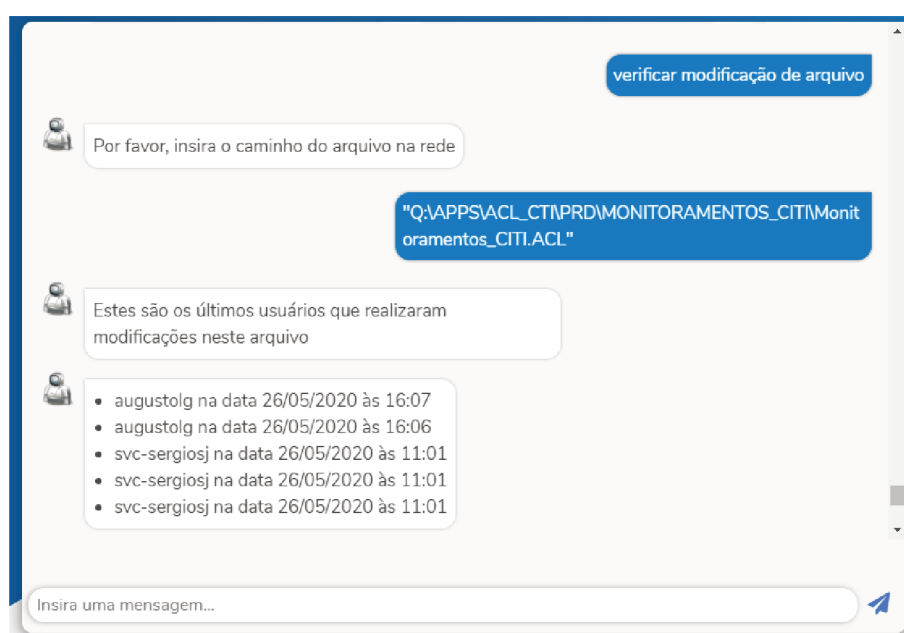
5.3.3 Modificações de arquivos

Grande parte dos arquivos digitais da empresa ficam armazenados em servidores que são acessados através de pastas compartilhadas nos computadores dos colaboradores. Desta maneira, os usuários conseguem realizar o compartilhamento de documentos com outras pessoas, facilitando o processo de edição e visualização destes arquivos.

As pastas compartilhadas possuem uma restrição de acesso para que somente os usuários alvos daquele diretório possam modificar os arquivos lá contidos. Algumas destas pastas possuem uma função habilitada que permite realizar a auditoria deste mesmo diretório. Desta maneira, é possível gerar logs com eventos relevantes de edição, deleção e criação de arquivos dentro daquela pasta. Estes arquivos são então indexados pelo Splunk, tornando possível realizar uma consulta para saber quais foram os usuários que alteraram algum arquivo dentro do diretório.

Baseado nesta possibilidade de consulta para verificação de modificações de arquivos, optou-se por implementar esta funcionalidade para que os usuários possam extrair este tipo de informação diretamente através do chatbot. Sendo assim, ao solicitar ao assistente uma consulta de quais foram os usuários que modificaram um arquivo recentemente, o usuário é indicado a inserir o caminho do arquivo na rede. Este caminho será utilizado pelo assistente para realizar a consulta no Splunk e, então, retornar ao usuário as últimas 5 pessoas que alteraram o arquivo alvo. A conversa exemplo na figura 26 demonstra uma extração completa deste tipo de informação.

Figura 26 – Conversa exemplo para verificação de modificações em arquivo da rede.



Para evitar o problema de o usuário não conseguir realizar a extração do caminho do arquivo que ele deseja realizar a verificação de alteração, foi criado um botão de sugestão específico que visa demonstrar de maneira simplificada como realizar este processo. Ao clicar neste botão, um vídeo explicativo é aberto demonstrando o passo a passo do procedimento de cópia do local do arquivo para um diretório exemplo. Ao finalizar o vídeo, o usuário é instruído a realizar o mesmo procedimento para o arquivo que ele deseja consultar.

5.3.4 Extração dos grupos do AD

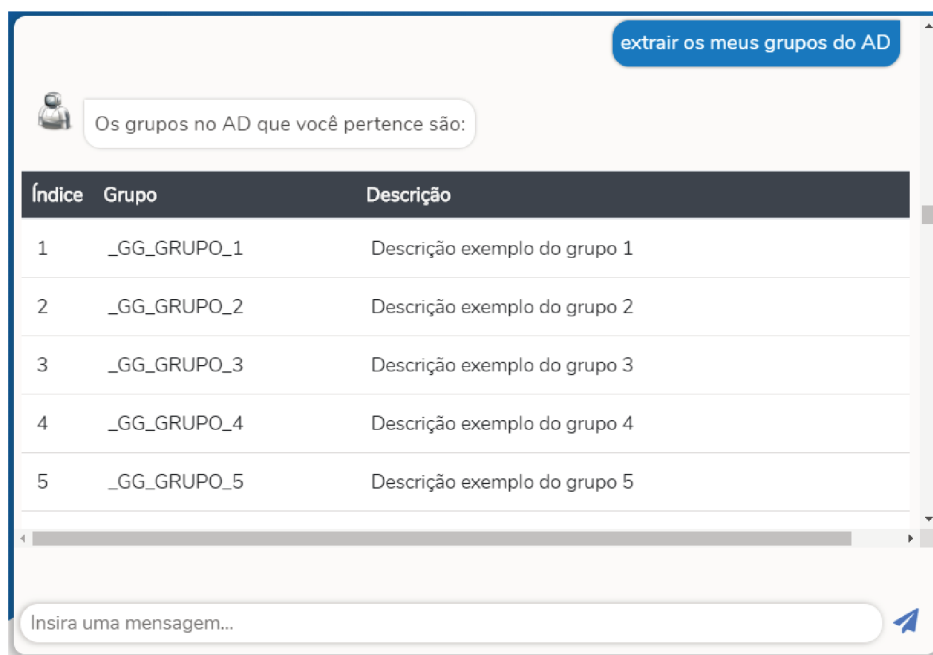
O *Active Directory* (AD) é uma ferramenta desenvolvida pela Microsoft para gerenciar recursos de uma rede de computadores. Através dele é possível criar usuários e grupos que permitem o gerenciamento de determinados recursos de uma infraestrutura de TI. Os grupos normalmente são estruturados para possibilitarem recursos como liberação de acesso à uma determinada ferramenta, concessão de acesso à uma pasta compartilhada, entre outros. É através do mecanismo de grupos que possibilita-se o gerenciamento de permissões de um usuário dentro da rede.

A funcionalidade do assistente virtual para extração de grupos do AD de um usuário permite que o colaborador tenha acesso rápido às suas permissões dentro da rede. É comum que os usuários não se recordem de terem acesso à uma funcionalidade liberada através de um grupo. Por este motivo, os colaboradores tendem a solicitar o acesso novamente ao mesmo grupo, gerando um aumento no número de solicitações da equipe de realiza o suporte de TI da empresa.

Para possibilitar a extração destas informações por meio do chatbot, é necessário realizar uma pesquisa dentro do AD. Esta pesquisa pode ser feita por meio do próprio Splunk, apesar de as informações não ficarem armazenadas nele. Isso só é possível, porque o Splunk consegue extrair os dados utilizando um protocolo de comunicação que permite interagir com o AD. Então, para extrair os dados solicitados pelo usuário, pode-se filtrar os grupos que ele possui acesso. Quando todos os grupos do usuário foram identificados, cria-se outra pesquisa para buscar o detalhamento de cada grupo. Este detalhamento permitirá ao usuário tomar conhecimento de qual acesso o grupo libera, já que somente através do nome, esta informação não é evidente.

Na figura 27, pode-se verificar o momento em que o usuário faz a solicitação dos grupos aos quais ele pertence e o assistente disponibiliza uma tabela com o nome e a descrição de cada grupo. Neste caso, os dados dos nomes dos grupos e os seus respectivos detalhamentos foram alterados para evitar a demonstração de informações particulares da empresa no documento.

Figura 27 – Conversa exemplo para extração de grupos do AD de um usuário.



Fonte – Elaborado pelo autor.

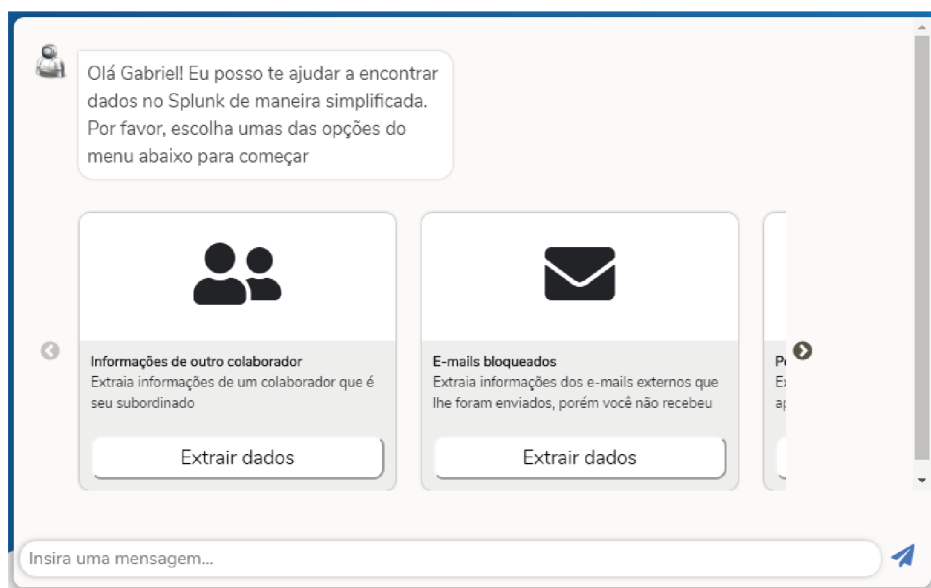
5.3.5 Extrações de dados de outros colaboradores

Os dados armazenados no Splunk são frequentemente solicitados por gestores e gerentes das equipes na empresa. Estes usuários desejam realizar consultas que envolvem os colaboradores das seções e departamentos que eles gerenciam, para que seja possível extrair informações relevantes dos colaboradores da área deles.

Para que seja possível extrair informações de outra pessoas é necessário que haja um controle de permissões de acessos. Este tipo de solicitação só deve estar disponível para colaboradores que trabalhem em cargos de gerência, onde existem outras pessoas sob seus controles.

Esta funcionalidade é implementada no assistente com uma checagem inicial do cargo do usuário que está utilizando a plataforma. É feita uma verificação através do banco de dados interno, com informações de todos os funcionários, para extrair qual é a função do colaborador na empresa atualmente. Para os usuários de gerência uma nova possibilidade de extração de dados é mostrada na tela inicial do assistente. Conforme a figura 28, um novo cartão de sugestão chamado de “Informações de outro colaborador” é mostrado ao usuário. Este mesmo cartão não fica disponível para um colaborador comum, assim como demonstrado nos exemplos anteriores deste capítulo. Sendo assim, o usuário do tipo gestor pode extrair, através deste cartão, dados de outros colaboradores que pertençam a mesma seção ou departamento dele.

Figura 28 – Interface inicial do assistente para gestores.

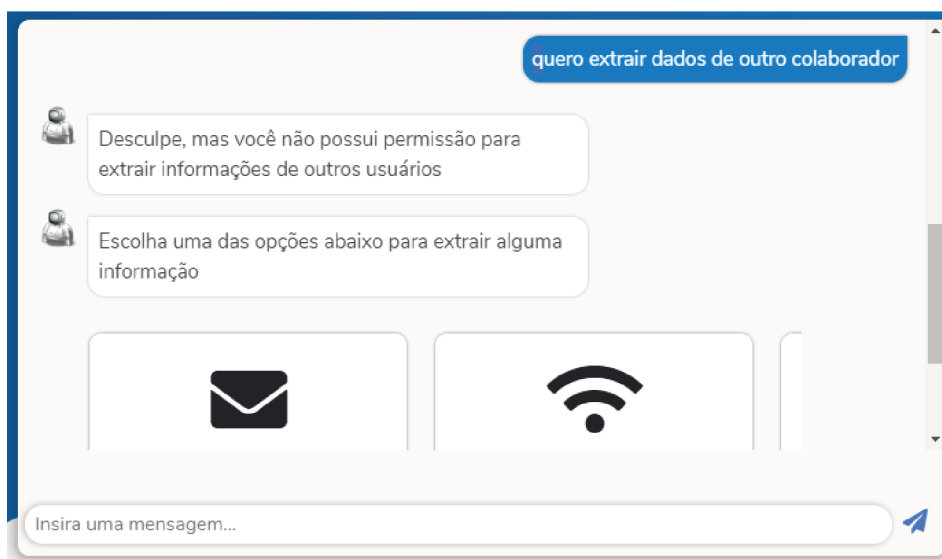


Fonte – Elaborado pelo autor.

Os tipos de dados que podem ser extraídos de outros colaboradores são limitados ao resumo dos e-mails trocados, ao perfil de acesso à internet e aos e-mails bloqueados. Ao solicitar ao chatbot uma extração de informações de outra pessoa, o usuário é questionado em relação ao e-mail do colaborador que ele deseja realizar a verificação. Neste momento o assistente realiza uma nova busca no banco de dados interno da empresa para verificar quais são os colaboradores que são subordinados àquele gestor. Logo em seguida, é criada uma lista com os e-mails destas pessoas para que o usuário do tipo gestor possa selecionar qual colaborador ele deseja extrair informações. Após a seleção do e-mail, é feita uma checagem para validar se o endereço inserido faz parte da lista de endereços das pessoas que estão sob a gerência do usuário que está utilizando o assistente. Caso a verificação retorne um resultado positivo, o usuário pode solicitar a extração de uma informação relacionada ao perfil do colaborador em questão.

Em casos em que o e-mail inserido pelo usuário (gestor) não esteja na listagem de permissões dele, uma mensagem é apresentada deixando claro que não é possível realizar a extração de informações. O mesmo é feito em casos em que um usuário comum (não gestor) tenta acessar esta funcionalidade e extrair informações de outro usuário, conforme a figura 29.

Figura 29 – Negação da extração de dados de outro colaborador



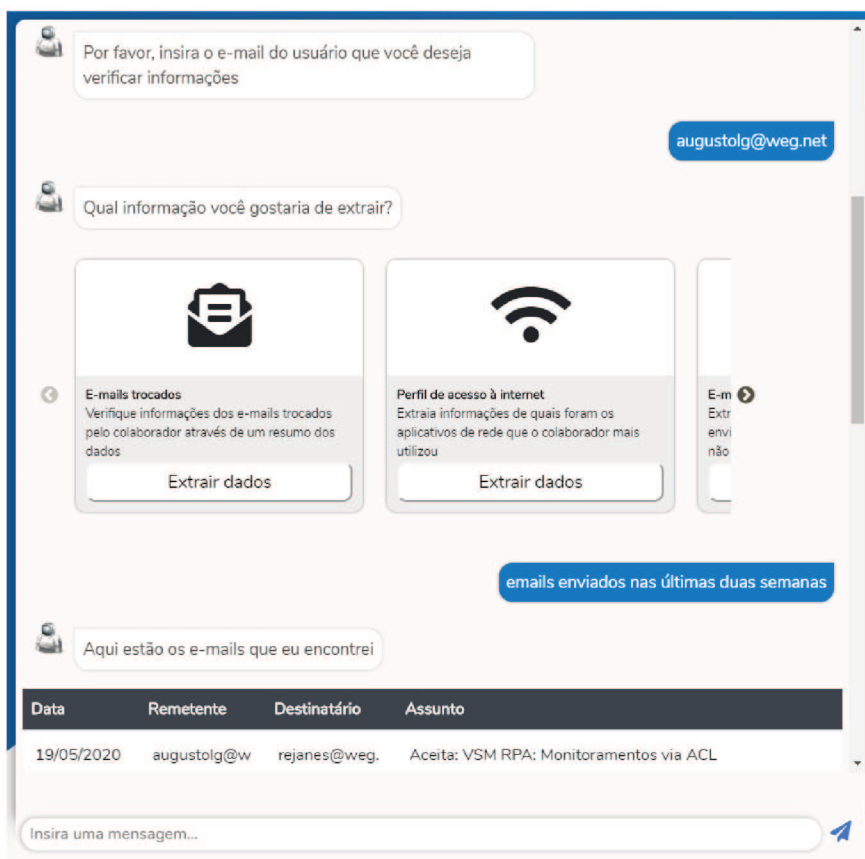
Fonte – Elaborado pelo autor.

5.3.6 E-mails trocados

Algumas das principais atividades de projetos e compartilhamento de informações acontecem através da troca de e-mails entre os colaboradores. Logicamente, cada colaborador consegue visualizar os seus e-mails enviados ou recebidos através da sua própria caixa de entrada. Entretanto, em algumas ocasiões, um gestor pode desejar visualizar informações dos e-mails dos colaboradores da sua área para tentar extrair dados como realização de tarefas, entregas de documento, entre outros.

Esta funcionalidade só pode ser utilizada através do menu dos gestores, devido à ambiguidade de um usuário ver informações dos seus próprios e-mails, conforme explicado anteriormente. Neste caso, o gestor pode inserir qual é o endereço de e-mail do colaborador que ele deseja verificar os e-mails trocados e logo em seguida o assistente solicita a inserção do período de extração. Assim que os dados são inseridos pelo gestor, o chatbot realiza a busca das informações e retorna os principais dados dos e-mails trocados por aquele usuário. Dentre estes dados estão informações como data, remetente, destinatário e assunto. O corpo dos e-mails não é mostrado por motivos de privacidade, evitando assim vazamentos de informações confidenciais. Assim, o gestor consegue ter uma noção das atividades realizadas por e-mail dos seus colaboradores e verificar se algumas tarefas que tinham sido planejadas foram de fato realizadas. No exemplo da figura 30, pode-se notar um gestor realizando uma extração dos e-mails enviados por um colaborador. Os resultados são apresentados em formato de tabela para facilitar a visualização das informações.

Figura 30 – Conversa exemplo para extração dos e-mails enviados por um colaborador da equipe do gestor



Fonte – Elaborado pelo autor.

6 ANÁLISE DE RESULTADOS

Uma análise quantitativa dos resultados do projeto foi realizada para gerar parâmetros de comparação com o objetivo de avaliar o desempenho do assistente virtual. Nesta seção são discutidas as análises das funcionalidades implementadas e das avaliações dos usuários.

6.1 USO DAS FUNCIONALIDADES

A análise de utilização de funcionalidades é importante para que os desenvolvedores do assistente virtual possam tomar conhecimento do perfil dos usuários e, com isso, passar a desenvolver novas aplicações com base nesse mesmo perfil. Desta forma, pode-se analisar se é válido manter alguma funcionalidade que não possui alta utilização e também pensar em expandir as capacidades das funcionalidades que são mais acessadas.

Para poder extrair os dados de utilização das funcionalidades do chatbot é preciso exportar informações de análise dentro da plataforma do Dialogflow. Através deste mecanismo, é possível verificar quais são as intenções que mais foram acionadas em um determinado período de tempo. Como as intenções representam as requisições dos usuários, pode-se extrapolar essa análise para gerar um gráfico que retrata as funcionalidades que mais foram requisitadas pelas pessoas que acessaram o chatbot. O limite de extração deste tipo de informação por meio do Dialogflow é de 30 dias. Como o assistente só foi disponibilizado para um grupo de usuários na parte final da execução do projeto, esta amostragem já é suficiente para entender o perfil de utilização dos colaboradores que tiveram a oportunidade de testar a ferramenta. O número de conversas analisadas neste período é de 331.

Analisando a tabela 1, nota-se que o recurso de extração de e-mails bloqueados foi o mais utilizado, contando com 41,7% das interações. Esse resultado demonstra que o fato de os usuários não receberem algum e-mail por motivo de bloqueio pode ser algo relevante para a continuidade das suas jornadas de trabalho. Por este motivo, a maioria dos colaboradores tendem a executar este recurso ao acessar o chatbot. Outro fator que pode ter influenciado nesta escolha é a posição desta funcionalidade nos cartões de sugestão. Este é o primeiro cartão de sugestão que aparece para o usuário, podendo influenciá-lo a clicar no botão do cartão para testar rapidamente o funcionamento do chatbot. Algo semelhante pode acontecer com o segundo recurso mais utilizado, com 17,3% da interações, que está relacionado com dados extraídos por usuários do tipo gestor. Para estes usuários, o recurso de extração de dados de outros colaboradores é o primeiro cartão de sugestão, podendo também influenciar na escolha. Apesar disso, neste recurso estão contidas todas as extrações de um gestor com relação aos colaboradores das suas áreas. Os gestores que realizaram

extrações de e-mails trocados, perfis de acessos de internet e e-mails bloqueados dos seus subordinados estão englobados nesta categoria. Portanto, o uso destes recursos demonstra também o interesse de extração de informações por parte dos usuários do tipo gestor.

A terceira funcionalidade mais utilizada foi a verificação de modificação de arquivos na rede. Mais de 16% dos usuários acessaram este recurso para tentar verificar alterações em algum arquivo que eles utilizam em suas rotinas de trabalho. Logo em seguida, o recurso mais recorrente foi a realização de nova consulta. Nestes casos o usuário extrai uma informação e posteriormente solicita ao chatbot a realização de uma nova pesquisa de dados.

É importante destacar ainda a categoria das interações não mapeadas, que por sua vez, foi ativada em 5,5% das conversas. Esta categoria é ativada quando o chatbot não possui uma resposta para a mensagem do usuário. Nestes casos, optou-se por guiar os usuários para as funções implementadas, através de uma mensagem padronizada e dos cartões de sugestão.

Tabela 1 – Dados de utilização das funcionalidades do assistente virtual.

Funcionalidade	Número de conversas	Percentual de conversas (%)
E-mails bloqueados	138	41,7
Extração de terceiros	57	17,3
Modificação de arquivos	53	16,1
Sites acessados	30	9,1
Nova consulta	18	5,5
Não mapeado	17	5,2
Outros	16	5,1

Fonte – Elaborado pelo autor.

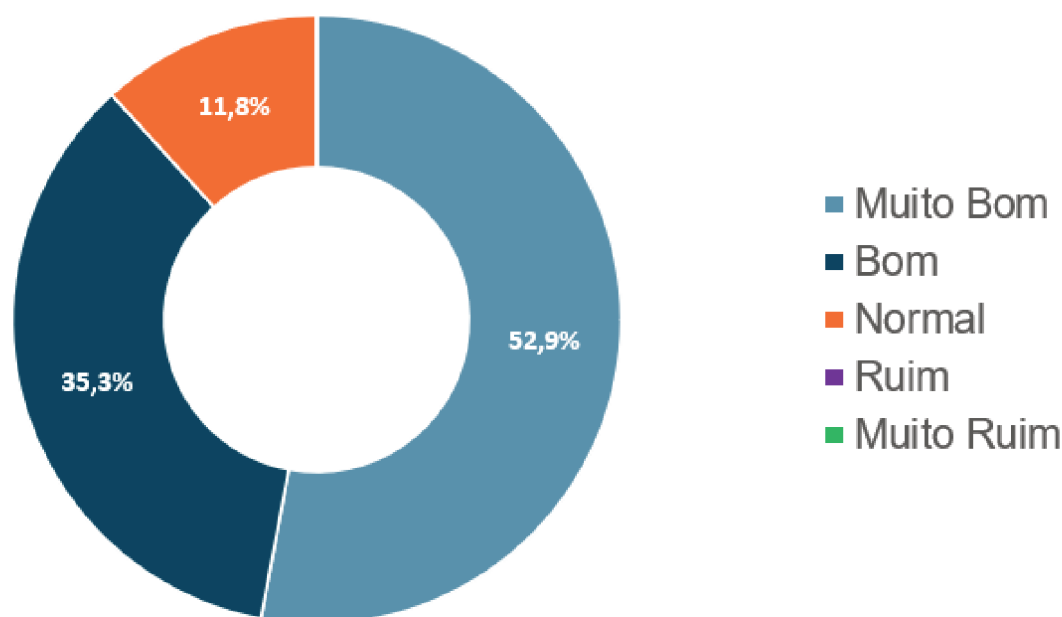
6.2 AVALIAÇÃO DAS FUNCIONALIDADES

Para analisar a experiência de utilização do assistente virtual, optou-se por criar um formulário de avaliação. Este formulário pode ser acessado pelo próprio chatbot durante a utilização da plataforma. Um botão de sugestão chamado de “Avaliar chatbot” aparece na tela do assistente toda vez que um usuário finaliza a extração de alguma informação. Desta maneira, caso o usuário opte por clicar no botão, uma breve pesquisa com quatro perguntas é apresentada na tela. O usuário pode então preencher os dados da pesquisa e, em seguida, retornar ao chatbot. Para o levantamento dos resultados destas pesquisas foram analisadas as respostas de 17 utilizadores da plataforma.

A primeira pergunta do formulário tem relação com a forma de comunicação do assistente. Nesta pergunta, busca-se entender se a maneira como o assistente se comunica agrada os usuários de maneira geral. Observando a figura 31, nota-se que a

grande maioria dos utilizadores aprovaram o formato de comunicação, avaliando este requisito com “Bom” ou “Muito bom”. Cerca de 11% dos usuários consideraram normal o formato de comunicação. Por tratar diretamente com colaboradores da empresa, o chatbot foi desenvolvido com uma linguagem formal. Além disso, o tipo de informação disponibilizada pelo assistente tornou propício este modo de comunicação. Através dos resultados da pesquisa, pôde-se verificar que o formato agradou os colaboradores.

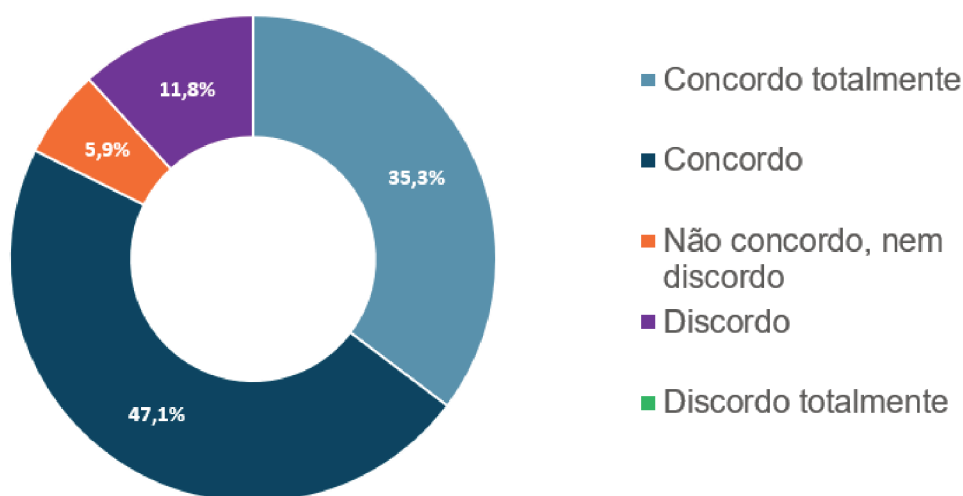
Figura 31 – Respostas dos usuários à pergunta: "Como você avalia a maneira de se comunicar do assistente?"



Fonte – Elaborado pelo autor.

A pergunta número dois tem relação com a possível otimização de tempo do usuário ao utilizar o chatbot. O resultado desta pergunta é relevante, já que foi um dos aspectos que foram levados em consideração para a criação deste projeto. Ao observar a figura 32, pode-se verificar que 35,3% das pessoas concordam totalmente com o fato de o assistente virtual ter otimizado o tempo delas. Outros 47,1% afirmaram que concordam com a pergunta. Estes números demonstram uma alta eficiência da extração de informações que foram propostas, agilizando o trabalho dos colaboradores ao utilizar o chatbot e atendendo os objetivos do projeto. Apesar disso, 11,8% das pessoas discordam da pergunta. Este fato pode ser causado pelo número de funcionalidades que foram implementadas. É possível que estes usuários desejassem extrair outros dados que não estavam disponíveis no momento em que testaram o assistente. Esse tipo de ocasião pode causar descontentamento por parte do usuário, levando-o a discordar da pergunta. Outros 5,6% dos usuários não concordaram, nem discordaram do questionamento.

Figura 32 – Respostas dos usuários à pergunta: "O assistente conseguiu otimizar o seu tempo através de alguma funcionalidade?"



Fonte – Elaborado pelo autor.

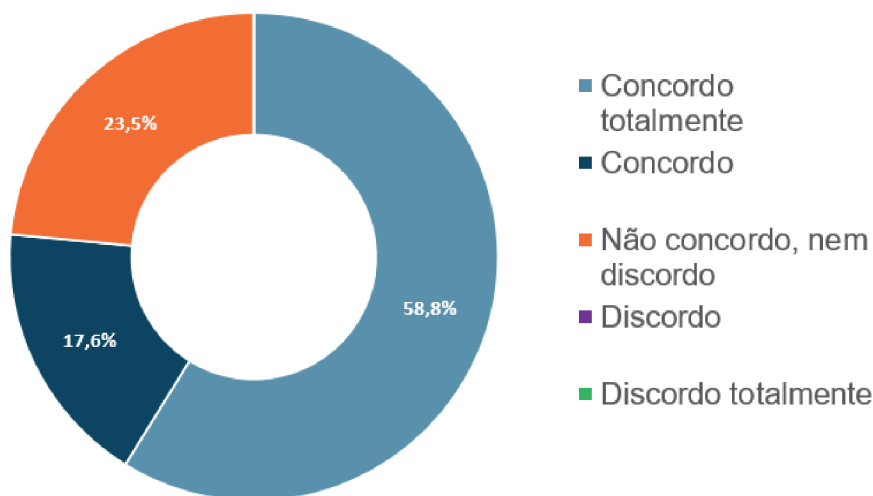
A terceira pergunta busca questionar o usuário com relação à simplificação de acesso à informação. Todos os dados disponibilizados pelo assistente somente podem ser extraídos por meio do Splunk. Por este motivo, os colaboradores não conseguiam ter acesso a eles de maneira simples. Era necessário solicitar o suporte de um outro colaborador especialista na plataforma para poder visualizar aquela informação.

Como pode ser visto na figura 33, 58,8% concordaram totalmente e 17,6% concordaram com a afirmação de simplificação de acesso à informação através do chatbot. Estes valores demonstram que a maior parte dos usuários conseguiram ter acesso aos dados do Splunk de maneira mais fácil do que anteriormente. Além disso, 23,5% dos usuários não concordaram, nem discordaram da pergunta. É possível que estes usuários não conseguiram acessar alguma informação que eles realmente procuravam, entretanto, obtiveram dados de outras funcionalidades que também se demonstraram úteis.

A última pergunta da pesquisa tem relação com a parte visual do chatbot. Nela, os usuários foram questionados com relação a qualidade da interface e das representações de informações através de tabelas, gráficos e mensagens de texto. Uma interface agradável faz com que os utilizadores de uma plataforma tendam a permanecer usufruindo dos seus recursos. Além disso, um formato de apresentação de informações bem configurado facilita a compreensão daqueles dados.

De acordo com os resultados da figura 34, 47,1% dos usuários classificaram como muito boa, 35,3% como boa e 17,6% como normal a qualidade gráfica do assistente virtual. Estes dados confirmam que os utilizadores gostaram do modo como as informações são apresentadas a eles quando interagem com o chatbot. Essa apro-

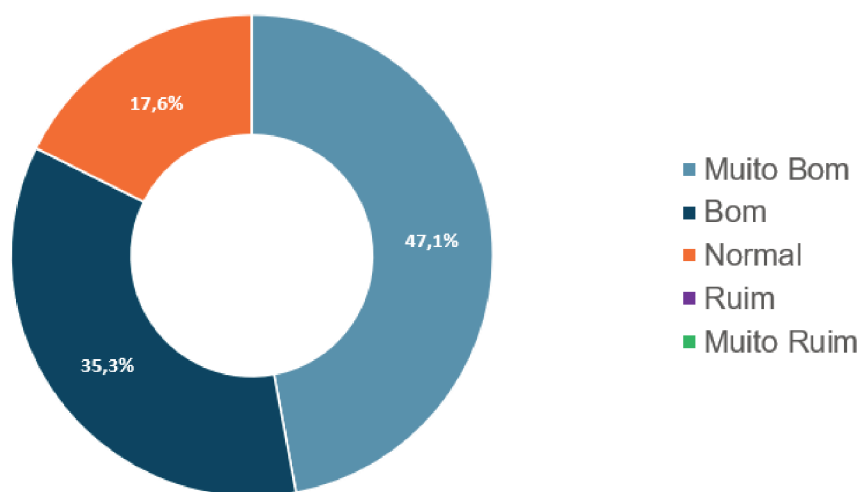
Figura 33 – Respostas dos usuários à pergunta: "O assistente conseguiu simplificar o acesso à alguma informação que você procurava?"



Fonte – Elaborado pelo autor.

vação pode ter uma relação direta com os contínuos feedbacks que foram solicitados aos colaboradores durante o desenvolvimento do projeto. Através de conversas com usuários foi possível implementar e modificar várias características visuais da interface durante o período final do desenvolvimento.

Figura 34 – Respostas dos usuários à pergunta: "Como você avalia a interface do assistente e o formato de apresentação de informações (gráficos, tabelas e texto)?"



Fonte – Elaborado pelo autor.

7 CONSIDERAÇÕES FINAIS E PERSPECTIVAS

Este projeto discute a criação e implementação de um assistente virtual para simplificar o acesso à informação contida em uma plataforma de SIEM. Anteriormente, para obter estes dados, os usuários necessitavam solicitar o suporte de colaboradores especializados na plataforma. A criação de um assistente do tipo chatbot visa facilitar essa cadeia de solicitações e fazer com que o usuário ganhe autonomia para realizar as suas próprias consultas, sem a necessidade de entrar em contato com outras pessoas. Com isso, objetiva-se diminuir o tempo para obtenção de informações e também a alocação exclusiva de colaboradores para este tipo de função.

A arquitetura proposta é composta por um *back-end* criado através do *framework* Flask. Por meio desta parte da arquitetura são realizadas as comunicações com o mecanismo de processamento de linguagem natural (Dialogflow) e também com a própria plataforma de SIEM (Splunk). O *back-end* é responsável por realizar o tratamento das mensagens do usuário e também por criar as pesquisas no Splunk que serão entregues aos usuários, conforme suas solicitações. Para facilitar a visualização e também melhorar a experiências de utilização do assistente virtual, foi desenvolvida uma interface gráfica com o conjunto de linguagens HTML, CSS e Javascript. Através desta interface, pode-se visualizar mensagens de texto trocadas entre o usuário e o assistente e também disponibilizar informações em formatos mais ricos visualmente, como por exemplo gráficos e tabelas. Além disso, o *front-end* permite a criação de cartões e botões de sugestões que orientam o usuário durante a utilização do assistente virtual, com o objetivo de facilitar o processo de obtenção de dados através da plataforma.

As funcionalidades disponíveis no assistente virtual foram desenvolvidas com base nas maiores demandas dos usuários com relação a extração de dados do Splunk na empresa. Com base em conversas com colaboradores anteriormente responsáveis por estas demandas, foram criados mecanismos para atender estas solicitações por meio do chatbot. Desta maneira, todas as funcionalidades implementadas já eram tratadas como solicitações recorrentes dentro do ambiente de TI da empresa.

Para avaliar o desempenho do projeto foi criado um questionário com perguntas generalistas relacionadas ao chatbot. Nele, são abordadas questões que visam entender se as funcionalidades implementadas foram capazes de suprir as expectativas dos desenvolvedores. Além disso, busca-se compreender se os mecanismos visuais fazem com que o usuário consiga interpretar as informações de maneira simplificada e efetiva. Em paralelo com este questionário, realiza-se uma análise das funcionalidades mais utilizadas pelos usuários que testaram o chatbot durante o período de avaliação deste projeto. Desta forma, pode-se fazer o levantamento dos recursos mais importantes e também propor eventuais novas funcionalidades com base no comportamento usuário.

Dentre as dificuldades encontradas no desenvolvimento do projeto pode-se citar as barreiras de acesso aos dados que são criadas devido às estruturas de segurança da informação dentro da empresa. Por este motivo foi necessário criar mecanismos capazes de se adequar a estas ferramentas para que fosse possível obter os dados necessários e em seguida disponibilizá-los aos usuários. Além disso, outra dificuldade relevante esteve presente na etapa de disponibilização da plataforma para os usuários finais. Para possibilitar o acesso dos usuários foi necessário criar um mecanismo de controle de autenticação e também transformar a aplicação em um container que poderia ser reutilizado em qualquer sistema operacional.

Os resultados obtidos no projeto foram considerados relevantes dentro do departamento da empresa. Pôde-se verificar que este tipo de solução apresenta um alto potencial para suprir as capacidades dos usuários, sem a necessidade de alocação exclusiva de colaboradores especialistas. Além disso, o fato de o usuário realizar todo o processo de maneira autônoma faz com que estas pessoas se sintam cada vez mais capazes e independentes.

A implementação de uma solução de chatbot demonstrou que este tipo de tecnologia pode se tornar uma ferramenta útil para solução de diversos outros problemas operacionais em empresas. O fato de tornar o chatbot menos abrangente foi considerado um dos fatores mais evidentes para que a solução obtivesse destaque. A criação de um chatbot focado em temas específicos fez com que o seu desempenho fosse superior aos chatbots generalistas, visto que, é possível aplicar regras muito mais restritas e que englobam a maior parte das possíveis interações com usuários.

7.1 SUGESTÕES DE TRABALHOS FUTUROS

Uma das limitações encontradas no projeto está no fato de que um usuário desenvolvedor especializado no chatbot precisa estar presente para a implementação de novas funcionalidades. Isto ocorre, porque o mecanismo de criação de novos recursos deve passar por três principais etapas: criação da intenção do usuário no mecanismo de processamento de linguagem natural, criação da pesquisa no Splunk e ainda no mapeamento destes recursos no *back-end* do chatbot. Estas etapas requerem que o desenvolvedor tenha conhecimento prévio específico destas três áreas para poder implementar uma nova funcionalidade.

Para tornar o assistente virtual mais complexo e mais abrangente sugere-se o desenvolvimento de um trabalho que permita facilitar a implementação de novas funcionalidades no assistente. Este mecanismo deve ser capaz de ser utilizado por um usuário com um nível de conhecimento baixo na arquitetura do chatbot para que mais pessoas possam passar a criar funcionalidades de acordo com as necessidades específicas das suas áreas de trabalho na empresa. Desta maneira, o assistente virtual iria suprir as necessidades de outros usuários, tornando-se mais abrangente.

REFERÊNCIAS

- ABUSHAWAR, Bayan; ATWELL, Eric. ALICE chatbot: trials and outputs. **Computación y Sistemas**, Centro de Investigación en Computación, IPN, v. 19, n. 4, p. 625–632, 2015.
- ADIWARDANA, Daniel *et al.* Towards a human-like open-domain chatbot. **arXiv preprint arXiv:2001.09977**, 2020.
- BARE, Jeff. **Amazon Lex – Build Conversational Voice Text Interfaces**. [S.l.: s.n.], nov. 2016. Disponível em: <https://aws.amazon.com/pt/blogs/aws/amazon-lex-build-conversational-voice-text-interfaces/>. Acesso em: 16 jun. 2020.
- BRANDTZAEG, Petter Bae; FØLSTAD, Asbjørn. Why people use chatbots. *In*: SPRINGER. INTERNATIONAL Conference on Internet Science. [S.l.: s.n.], 2017. P. 377–392.
- CAMBRIA, Erik; WHITE, Bebo. Jumping NLP curves: A review of natural language processing research. **IEEE Computational intelligence magazine**, IEEE, v. 9, n. 2, p. 48–57, 2014.
- CANONICO, Massimo; DE RUSSIS, Luigi. A comparison and critique of natural language understanding tools. **Cloud Computing**, v. 2018, p. 120, 2018.
- CARASSO, David. **Quick Splunk Reference for SQL Users**. [S.l.: s.n.], out. 2010. Disponível em: https://www.splunk.com/en_us/blog/tips-and-tricks/splunk-for-sql-users.html. Acesso em: 24 mai. 2020.
- CHOWDHURY, Gobinda G. Natural language processing. **Annual review of information science and technology**, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003.
- DALE, Robert. The return of the chatbots. **Natural Language Engineering**, Cambridge University Press, v. 22, n. 5, p. 811–817, 2016.
- DUCKETT, Jon. **Beginning html, xhtml, css, and javascript**. [S.l.]: John Wiley & Sons, 2011.
- FUCUCHIMA, Letícia. **WEG avança com oferta de soluções digitais à indústria**. [S.l.: s.n.], jul. 2020. Disponível em: <https://valor.globo.com/empresas/noticia/2020/07/10/weg-avanca-com-oferta-de-solucoes-digitais-a-industria.ghtml>. Acesso em: 10 jul. 2020.
- GENESERETH, Michael R; NILSSON, Nils J. **Logical foundations of artificial intelligence**. [S.l.]: Morgan Kaufmann, 2012.

- GOASDUFF, Laurence. **Chatbots Will Appeal to Modern Workers**. [S.l.: s.n.], jul. 2019. Disponível em: <https://www.gartner.com/smarterwithgartner/chatbots-will-appeal-to-modern-workers/>. Acesso em: 12 jun. 2020.
- GOOGLE. **Dialogflow Documentation | Google Cloud**. [S.l.]: Google. Disponível em: <https://cloud.google.com/dialogflow/docs/>. Acesso em: 6 jun. 2020.
- GUNKEL, David J. Comunicação e inteligência artificial: novos desafios e oportunidades para a pesquisa em comunicação. **Galáxia. Revista do Programa de Pós-Graduação em Comunicação e Semiótica. ISSN 1982-2553**, n. 34, 2017.
- HIGASHINAKA, Ryuichiro *et al.* Towards an open-domain conversational system fully based on natural language processing. *In: PROCEEDINGS of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. [S.l.: s.n.], 2014. P. 928–939.
- HOPE, Computer. **What is a Markup Language?** [S.l.: s.n.], jun. 2019. Disponível em: <https://www.computerhope.com/jargon/m/markup-language.htm>. Acesso em: 2 jun. 2020.
- IBM. **Assistente do Watson**. [S.l.]: IBM. Disponível em: <https://cloud.ibm.com/docs/assistant?topic=assistant-index>. Acesso em: 6 jun. 2020.
- LIDDY, Elizabeth D. *Natural language processing*, 2001.
- MICROSOFT. **Assistente do Watson**. [S.l.]: Microsoft. Disponível em: <https://azure.microsoft.com/pt-br/services/cognitive-services/language-understanding-intelligent-service/>. Acesso em: 6 jun. 2020.
- MOOR, James. The Dartmouth College artificial intelligence conference: The next fifty years. **Ai Magazine**, v. 27, n. 4, p. 87–87, 2006.
- NILSSON, Nils J; NILSSON, Nils Johan. **Artificial intelligence: a new synthesis**. [S.l.]: Morgan Kaufmann, 1998.
- PETERS, Tim. Pep 20—the zen of python. **Python Enhancement Proposal**, 2004.
- PRATT, Mary K. **What is SIEM software? How it works and how to choose the right tool**. [S.l.]: CSO, nov. 2017. Disponível em: <https://www.csoonline.com/article/2124604/what-is-siem-software-how-it-works-and-how-to-choose-the-right-tool.html>. Acesso em: 10 jun. 2020.
- RAHIM, Siti Rohaya Mat *et al.* Artificial intelligence, smart contract and islamic finance. **Asian Soc Sci**, v. 14, n. 2, p. 145, 2018.

REITMAN, Walter Ralph. **Artificial intelligence applications for business: proceedings of the NYU symposium, May, 1983**. [S.l.]: Intellect Books, 1984.

ROLLER, Stephen *et al.* Recipes for building an open-domain chatbot. **arXiv preprint arXiv:2004.13637**, 2020.

SHAWAR, Bayan Abu; ATWELL, Eric. Chatbots: are they really useful? *In*: 1. LDV forum. [S.l.: s.n.], 2007. P. 29–49.

SOMMERVILLE, Ian. **Engenharia de Software. ed.** [S.l.]: São Paulo, SP: Pearson Prentice Hall, 2011. P. 59.

TRAMUNT IBAÑOS, Ana; BATISTA PAIL, Daisy. **Fundamentos linguísticos e computação**. [S.l.]: EDIPUCRS, 2017.

TURING, Alan. Computação e inteligência. **FC Hansem, Trad. Em J. de F. Teixeira (Org.), Cérebros Máquinas e Consciência: Uma Introdução à Ciência da Mente. São Carlos: Editora da UFSCar.(Trabalho original publicado em 1950)**, 1996.

UNIVERSITY, American. **5 Ways Technology Has Changed the Communication Field**. [S.l.: s.n.], fev. 2020. Disponível em:
<https://programs.online.american.edu/msc/masters-strategic-communication/resources/5-ways-technology-has-changed-the-communication-field>. Acesso em: 12 jun. 2020.

VAISH, Aakrit. **Five Reasons Why Chatbots are the Future of Customer Service**. [S.l.: s.n.], jan. 2019. Disponível em:
<https://www.entrepreneur.com/article/325830>. Acesso em: 6 jun. 2020.

VALENTE, Marco Tulio. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. [S.l.: s.n.], 2020.

VINYALS, Oriol; LE, Quoc. A neural conversational model. **arXiv preprint arXiv:1506.05869**, 2015.

WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. [S.l.]: Elsevier Editora Ltda., 2013.

WEERTS, IJM. **The detection and handling of verbal irony in human-computer interaction: a comparison between four chatbots**. 2018. B.S. thesis.

WEG. **Perfil Corporativo**. [S.l.: s.n.], jan. 2020. Disponível em:
<https://ri.weg.net/a-weg/perfil-corporativo/#:~:text=Fundada%20em%201961,%20a%20WEG,%20minera%C3%A7%C3%A3o,%20entre%20muitos%20outros..> Acesso em: 11 jul. 2020.

WEIZENBAUM, Joseph. ELIZA—a computer program for the study of natural language communication between man and machine. **Communications of the ACM**, ACM New York, NY, USA, v. 9, n. 1, p. 36–45, 1966.

WIT. **Build Natural Language Experiences**. [S.l.: s.n.]. Disponível em: <https://wit.ai/>. Acesso em: 12 jun. 2020.

WORSWICK, Steve. **Mitsuku wins Loebner Prize 2018!** [S.l.]: pandorabots-blog, set. 2018. Disponível em: <https://medium.com/pandorabots-blog/mitsuku-wins-loebner-prize-2018-3e8d98c5f2a7>. Acesso em: 6 jun. 2020.