

Universidade Federal de Santa Catarina

Thiago Pereira da Silva

Desenvolvimento de um sistema do tipo *Chatbot* para o curso de
Sistemas de Informação

Florianópolis

2021

Thiago Pereira da Silva

Desenvolvimento de um sistema do tipo *Chatbot* para o curso
de Sistemas de Informação

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Maicon Rafael Zatelli

Florianópolis, abril de 2021

Resumo

A interação entre humano e máquina está cada vez mais frequente nos dias de hoje, com isso o fornecimento e troca de informação está sendo cada vez mais automatizado. No entanto, a comunicação de um usuário com uma máquina através da linguagem natural ainda encontra diversos desafios. Além disso, existem diversas bibliotecas computacionais de conversação humano-máquina, porém, em sua grande maioria são bibliotecas para serem incorporadas como uma parte do sistema, ou seja, são necessárias diversas camadas de entorno até que o sistema possa ser utilizado por usuários na *web* e gerenciado por um administrador de forma amigável. Dessa forma, o objetivo do presente trabalho foi desenvolver um sistema do tipo Chatbot que apresente a solução completa, desde o processamento de linguagem natural adaptado à língua portuguesa, utilizando uma biblioteca de código aberto, até a interface de bate-papo e o sistema administrativo para gerenciamento do Chatbot. O contexto da conversação do Chatbot é no âmbito do curso de graduação em Sistemas de Informação da Universidade Federal de Santa Catarina. Após o desenvolvimento do sistema, o Chatbot foi exposto aos alunos do curso, obtendo um total de 394 perguntas em 83 sessões únicas. Dentre as respostas enviadas pelos Chatbot, 51,7% obtiveram uma avaliação positiva. Portanto, o trabalho atingiu o objetivo criando um sistema completo de cadastro de perguntas e respostas, treinamento e interação com usuário em formato de bate-papo.

Palavras-chave: Chatbot, Linguagem natural, Conversação, Código-livre.

Lista de ilustrações

Figura 1 – Pesquisa pelo termo 'chatbot' no Google Trends.	19
Figura 2 – Serviços que compõe o sistema proposto no trabalho.	31
Figura 3 – Serviços do sistema Chatbot isolados em Contêineres.	33
Figura 4 – Representação visual das entidades do banco de dados do sistema através do diagrama ER.	35
Figura 5 – Diagrama de atividade UML relacionado à interação do usuário com o Chatbot.	36
Figura 6 – Diagrama de atividade UML relacionado à ação de treinamento realizada pelo administrador.	37
Figura 7 – Biblioteca Python Chatterbot.	39
Figura 8 – Interface <i>web</i> para interação do usuário com o Chatbot.	44
Figura 9 – Interface <i>web</i> do Chatbot com a caixa de bate-papo (<i>chat</i>) aberta.	44
Figura 10 – Interface <i>web</i> do Chatbot com formulário de envio de mensagem inativo aguardando o <i>feedback</i> do usuário.	45
Figura 11 – Interface <i>web</i> para interação do usuário com o Chatbot.	45
Figura 12 – Interface <i>web</i> do Chatbot com a caixa de avaliação final aberta.	46
Figura 13 – Chatbot Frontend acessado via <i>Smartphone</i>	46
Figura 14 – Interface <i>web</i> de autenticação para acesso à área administrativa.	47
Figura 15 – Página inicial da interface <i>web</i> da área administrativa.	48
Figura 16 – Funcionalidades da interface <i>web</i>	49
Figura 17 – Interface <i>web</i> - cadastro de novos conhecimentos.	50
Figura 18 – Área de visualização de <i>feedbacks</i> para as respostas fornecidas pelo Chatbot.	51
Figura 19 – Funcionalidade de pesquisa de mensagens entre um usuário e o Chatbot.	52
Figura 20 – Área de visualização das avaliações finais de interação com o Chatbot.	52
Figura 21 – Avaliações finais registradas pelos usuários.	54

Lista de tabelas

Tabela 1 – Comparação entre Chatbots de diferentes cenários.	24
Tabela 2 – Exemplo de treinamento e teste da biblioteca Python <i>Chatterbot</i>	40
Tabela 3 – Total de perguntas e <i>feedbacks</i> negativos e positivos realizados no Chatbot.	53
Tabela 4 – Assuntos questionados ao Chatbot.	55

Lista de abreviaturas e siglas

<i>API</i>	<i>Application Programming Interface</i>
<i>CTC</i>	Centro Tecnológico
<i>FAQ</i>	<i>Frequently Asked Questions</i>
<i>HTML</i>	<i>HyperText Markup Language</i>
<i>IA</i>	Inteligência Artificial
<i>INE</i>	Informática e Estatística
<i>JSON</i>	<i>JavaScript Object Notation</i>
<i>LN</i>	Linguagem Natural
<i>NLTK</i>	<i>Natural Language Toolkit</i>
<i>ORM</i>	<i>Object Relational Mapper</i>
<i>PDF</i>	<i>Portable Document Format</i>
<i>PLN</i>	Processamento Linguagem Natural
<i>REST</i>	<i>Representational State Transfer</i>
<i>SGBD</i>	Sistema de Gerenciamento de Banco de Dados
<i>SQL</i>	<i>Structured Query Language</i>
<i>UFSC</i>	Universidade Federal de Santa Catarina
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>YAML</i>	<i>YAML Ain't Markup Language</i>

Sumário

1	Introdução	13
1.1	Objetivos	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	15
1.2	Escopo do trabalho	16
1.3	Metodologia	16
1.3.1	Revisão bibliográfica e contextualização sobre Chatbots	17
1.3.2	Levantamento dos requisitos necessários para o funcionamento do sistema proposto	17
1.3.3	Análise de Chatbots e bibliotecas de código livre disponíveis para desenvolvimento do sistema proposto	17
1.3.4	Definição da arquitetura e tecnologias envolvidas em cada módulo do sistema	17
1.3.5	Validação do Chatbot através da interação e <i>feedback</i> de usuários	17
1.3.6	Análise dos resultados alcançados através das avaliações dos usuários	18
1.3.7	Proposições para trabalhos futuros	18
2	Fundamentação Teórica	19
2.1	Chatbot	19
2.2	Processamento da Linguagem Natural (PLN)	20
2.3	Aprendizado de Máquina	21
3	Trabalhos Relacionados	23
4	Desenvolvimento	27
4.1	Especificação dos Requisitos de <i>Software</i>	27
4.1.1	Requisitos Funcionais	27
4.1.2	Requisitos Não Funcionais	28
4.1.3	Regras de Negócio	29
4.2	Arquitetura do Sistema	31
4.2.1	Serviço Chatbot API	31
4.2.2	Serviço Chatbot Frontend	32
4.2.3	Serviço Chatbot Admin	32
4.2.4	Persistência de dados	32
4.2.5	Orquestração do sistema via Docker Compose	33
4.3	Modelagem	34
4.3.1	Diagrama de Entidade Relacionamento	34
4.3.2	Diagrama de Atividades	35
4.4	Biblioteca Python ChatterBot	37

4.4.1	Funcionamento da biblioteca Python ChatterBot	38
4.5	Implementação	40
4.5.1	Serviço Chatbot API	41
4.5.2	Serviço Chatbot Frontend	43
4.5.3	Serviço Chatbot Admin	47
5	Resultados e Discussões	53
6	Conclusão	57
6.1	Trabalhos futuros	58
Referências		61
Apêndices		65
APÊNDICE A	Arquivo YAML com as instruções para o orquestrador de contê- neres Docker compose.	67
APÊNDICE B	Código Dockerfile desenvolvido para a imagem chatterbot:1.1.0a7.	71
APÊNDICE C	Artigo	73

1 Introdução

Em um mundo onde os meios de comunicação digital têm crescido rapidamente, ferramentas de automação têm se tornado um diferencial no mercado ao tornar os canais de comunicação mais eficientes. A interação entre o ser humano e o computador pode ocorrer de diversas formas, entre elas, a comunicação por linguagem natural (LN). A primeira aplicação computacional desenvolvida com o objetivo de simular uma conversa foi o software ELIZA, criado por Joseph Weizenbaum entre 1964 e 1966 (WEIZENBAUM et al., 1966). Desde então, os softwares de conversa têm avançado muito, diariamente é possível entrar em contato com assistentes do tipo Chatbot em computadores ou celulares, como é o caso da Siri da Apple, Google Assistente, Alexa da Amazon, Cortana da Microsoft, Zenbot, Robô Ed, Cedrico, entre outros.

O termo Chatbot (ou Chatterbot) vem do inglês, onde *chatter* significa "conversa fiada" e *bot* é uma abreviatura de *robot*, robô em português, consistindo em uma aplicação que proporciona um diálogo entre uma máquina e um ser humano. Esse tipo de aplicação têm sido largamente utilizada em canais de comunicação. Segundo Rahane et al. (2018), chatbots podem ser úteis em fornecer serviços a clientes, apresentar recomendações de produtos, atrair consumidores para diferentes campanhas de marketing, além de muitas outras áreas de negócios.

Para que um diálogo entre homem e máquina seja possível é necessário que a aplicação interprete os dados de entrada e realize um processamento a fim de identificar a melhor resposta para o usuário. Essa tecnologia pode ser utilizada em qualquer tipo de âmbito institucional em que haja necessidade de fornecimento e/ou recebimento de informação, a qual possa ser automatizada com objetivos específicos.

O termo Inteligência Artificial foi cunhado na década de 50 por um cientista da computação chamado John McCarthy como "a ciência e engenharia de máquinas inteligentes, especialmente programas de computadores" (NEUHAUSER; KREPS, 2011), tecnologia a qual vêm sendo aplicada atualmente em diversas áreas. A IA pode ser entendida como um sistema não natural que procura mimetizar funções cognitivas humanas, como por exemplo, o aprendizado. Diversas tentativas de utilização de IA no mercado até a década de 90 tiveram resultados limitados, porém, com a disseminação da internet, os sistemas de IA começaram a se tornar cada vez mais presentes. Os principais sistemas que se beneficiaram com a IA foram os mecanismos de busca e os sistemas de recomendação. Hoje em dia pode-se observar aplicações utilizando IA em praticamente todas as áreas de negócios e industriais, seja nos diversos tipos de atendimento ao cliente, sistemas de recomendações em qualquer tipo de aplicação mobile ou smartTV, assistentes virtuais,

entre outros.

Quando se trata de interação, por meio do diálogo, entre um ser humano e uma máquina, o principal desafio da máquina é compreender a intenção da mensagem do ser humano. Segundo Russel e Norvig (2013), a compreensão da linguagem exige a compreensão de dois aspectos, o assunto e o contexto da mensagem e não apenas a compreensão da estrutura das frases. Atualmente, existem diversas empresas que fornecem plataformas para construção de aplicações de diálogo com usuários humanos, como é o caso da IBM Watson¹, Microsoft bot², Dialogflow³, entre outras. No entanto, o principal idioma utilizado nessas ferramentas é o inglês e, na maioria dos casos, a tradução automatizada para a língua portuguesa encontra dificuldades, pois o contexto da conversa pode ser diferente da tradução literal das palavras ou pequenas frases. Segundo Rahman, Mamun e Islam (2017), os Chatbots ainda carecem de uma vasta lógica de recursos linguísticos. Consultas complexas ainda fornecem problemas de formas singulares e plurais, sinônimos e análise sentimental.

Aplicações de Chatbot também são utilizadas em portais *web* com grande quantidade de informação. A navegação por menus e diferentes níveis de páginas pode dificultar a busca por conteúdos do portal. Desta forma, uma pequena caixa de bate-papo com um Chatbot na página principal de um portal pode ter um ganho de tempo muito grande para o usuário, uma vez que o Chatbot pode responder a dúvida com uma mensagem completa ou uma URL apontando para a localização correta da informação. O portal do curso de Sistemas de Informação da UFSC (<https://sin.ufsc.br>) possui uma grande quantidade de informações, apesar da boa organização, um aplicativo de Chatbot na página inicial poderia fornecer ainda mais agilidade aos estudantes e demais usuários. Além de poder fornecer informações de forma ágil de diferentes níveis do portal, também pode conter informações que estão apenas disponíveis normalmente em arquivos do tipo PDF, documentos e até informações não disponíveis no portal.

Dessa forma, o presente trabalho pretende desenvolver um sistema de Chatbot com base em um mecanismo de conversação de código aberto. Este sistema deve permitir um fácil gerenciamento da base de conhecimento do Chatbot, com a possibilidade incrementar o seu banco de dados com novas informações. Além disso, o administrador será capaz de supervisionar diálogos não compreendidos pelo Chatbot, adicionando respostas às perguntas não reconhecidas visualizadas no histórico de conversas, com o objetivo de supervisionar e ensinar novos conhecimentos ao sistema. O tema empregado ao conhecimento do Chatbot é relacionado ao curso de graduação em Sistemas de Informação da Universidade Federal de Santa Catarina, nesse contexto, os usuários poderão dialogar com o Chatbot a fim de obter informações de forma rápida e interativa sobre dúvidas e questionamentos usuais, sem a

¹ <https://www.ibm.com/watson>

² <https://azure.microsoft.com/en-us/services/bot-service>

³ <https://cloud.google.com/dialogflow>

necessidade de uma pesquisa mais aprofundada no portal ou contato com coordenador e chefe de expediente.

1.1 Objetivos

1.1.1 Objetivo Geral

O presente trabalho tem o objetivo de desenvolver uma sistema *web* do tipo *chatbot* com uma base de conhecimento relacionada ao curso de Sistemas de Informação da UFSC. Este sistema engloba uma aplicação para processamento de linguagem natural adaptado à língua portuguesa, uma interface de conversação e uma área administrativa.

1.1.2 Objetivos Específicos

Considerando o desenvolvimento do trabalho e o objetivo geral apresentado, destacam-se os seguintes objetivos específicos:

- Especificar um conjunto de requisitos do sistema;
- Realizar um levantamento de tecnologias *web* e tecnologias relacionadas à *chatbots* para seleção de bibliotecas *open source*, plausíveis para compor o sistema do trabalho;
- Desenvolvimento de uma aplicação, com base em bibliotecas de código aberto, para interpretar a linguagem natural com fluxo básico de um *chatbot*: entrada, processamento e saída;
- Desenvolvimento de uma interface *web* para conversação entre o usuário e o *chatbot*;
- Desenvolvimento de um sistema administrativo para inserção de diálogos para aumento da base de conhecimento do *chatbot*;
- Inserção de informações sobre o curso de Sistemas de Informação disponibilizadas no portal do Departamento de Informática e Estatística e fornecidas através da secretaria e coordenação do curso;
- Avaliação dos resultados através da verificação da eficácia na resolução das perguntas realizadas pelos usuários.
- Realizar um levantamento dos assuntos mais questionados pelos usuários que interagiram com o Chatbot.

1.2 Escopo do trabalho

O presente trabalho restringe-se ao desenvolvimento de um sistema do tipo Chatbot com o objetivo de simular uma conversa entre pessoas. As respostas fornecidas pelo Chatbot são consultadas em uma base de conhecimento previamente cadastrada e com a possibilidade de incrementação. Toda a base de conhecimento estará contextualizada em informações relacionadas apenas ao Curso de Sistemas de Informação da Universidade Federal de Santa Catarina (UFSC).

O crescimento da base de conhecimento não se fará automaticamente pela conversação com os usuários. Será necessária a intervenção administrativa para cadastramento das respostas sobre as questões não resolvidas pelo Chatbot.

Em relação às tecnologias utilizadas, está excluída a utilização de tecnologias privadas, sendo assim, as partes que compõe o sistema utilizarão bibliotecas de código aberto, assim como as linguagens de programação escolhidas para cada módulo do sistema.

1.3 Metodologia

Com o objetivo de facilitar o acesso às informações do curso de Sistemas de Informação da UFSC, este trabalho propõe a construção de um sistema de Chatbot. Este sistema buscará agilizar e facilitar o acesso às informações pertinentes ao curso de Sistemas de Informação. Para atingir os objetivos propostos, o presente trabalho utilizou os procedimentos metodológicos abaixo em ordem cronológica:

- Revisão bibliográfica e contextualização sobre Chatbots;
- Levantamento dos requisitos necessários para o funcionamento do sistema proposto;
- Análise de Chatbots e bibliotecas de código livre disponíveis para desenvolvimento do sistema proposto;
- Definição da arquitetura e tecnologias envolvidas em cada módulo do sistema;
- Desenvolvimento e implementação de cada módulo do sistema;
- Validação do Chatbot através da interação e *feedback* de usuários;
- Análise dos resultados alcançados através das avaliações dos usuários;
- Proposições para trabalhos futuros.

1.3.1 Revisão bibliográfica e contextualização sobre Chatbots

Conceituar o termo Chatbot e contextualizar seu entorno em relação às bibliotecas de código aberto disponíveis na literatura, formas de processamento de linguagem natural, bem como conceitos importantes que cercam este tipo de sistema.

1.3.2 Levantamento dos requisitos necessários para o funcionamento do sistema proposto

Nesta etapa é executado um levantamento dos requisitos funcionais, não funcionais e regras de negócio para que o leitor compreenda o funcionamento do sistema e, principalmente, quais as suas características. A representação através de diagramas também foi utilizada para mostrar o funcionamento básico do sistema.

1.3.3 Análise de Chatbots e bibliotecas de código livre disponíveis para desenvolvimento do sistema proposto

Nesta etapa são analisados Chatbots existentes, suas funções, idiomas e bibliotecas utilizadas para o desenvolvimento, através do levantamento das principais características, auxiliando na decisão da melhor biblioteca para o sistema proposto.

1.3.4 Definição da arquitetura e tecnologias envolvidas em cada módulo do sistema

Nesta etapa são definidas a arquitetura do sistema e tecnologias envolvidas com base nos requisitos elencados anteriormente. Cada serviço (módulo) é descrito detalhadamente para auxiliar na compreensão das características e propriedades de cada componente do sistema.

1.3.5 Validação do Chatbot através da interação e feedback de usuários

Nesta etapa, com o sistema desenvolvido, realiza-se a exposição da interface de conversação para alunos do curso de Sistemas de Informação realizarem interações. A validação é realizada através da análise manual de perguntas e respostas, verificando se as respostas são coerentes com as perguntas fornecidas pelos usuários. Além disso, também serão verificados os *feedbacks* fornecidos pelos usuários para as respostas enviadas pelo Chatbot, validando assim a capacidade de comunicação do Chatbot.

1.3.6 Análise dos resultados alcançados através das avaliações dos usuários

Através dos *feedbacks* dos usuários e de uma avaliação final da interação com o Chatbot, serão avaliados os resultados alcançados com a primeira interação com usuários finais. Além disso, a análise dos resultados consiste também em realizar um levantamento dos principais assuntos abordados pelos usuários.

1.3.7 Proposições para trabalhos futuros

Durante as etapas de desenvolvimento, implementação e avaliação dos resultados, as oportunidades de melhoria que não foram incorporadas no presente trabalho são descritas como propostas para trabalhos futuros. Essas oportunidades de melhoria estão baseadas em dificuldades, problemas encontrados e sugestões de avanço ou crescimento das aplicações que possam melhorar a qualidade do sistema.

2 Fundamentação Teórica

Este capítulo apresenta uma revisão teórica dos conceitos fundamentais para a compreensão e desenvolvimento do presente trabalho. Além de fornecer uma contextualização do presente trabalho em meio ao estado de conhecimento da literatura sobre o tema proposto.

2.1 Chatbot

Os termos "chatbot", "chatterbot" ou simplesmente "bot", podem ser definidos como um programa de computador com o objetivo de tentar simular um ser humano em uma conversa com um usuário humano. Para que essa conversa ocorra, a forma de interação é através da linguagem natural ¹ que, ao contrário da linguagem formal² não pode ser diretamente traduzida em operações matemáticas precisas, mas pode ter informações e instruções extraídas, processadas e respondidas na mesma linguagem, sendo esse mecanismo de diálogo denominado de *chatbot* (HAPKE; LANE; HOWARD, 2019).

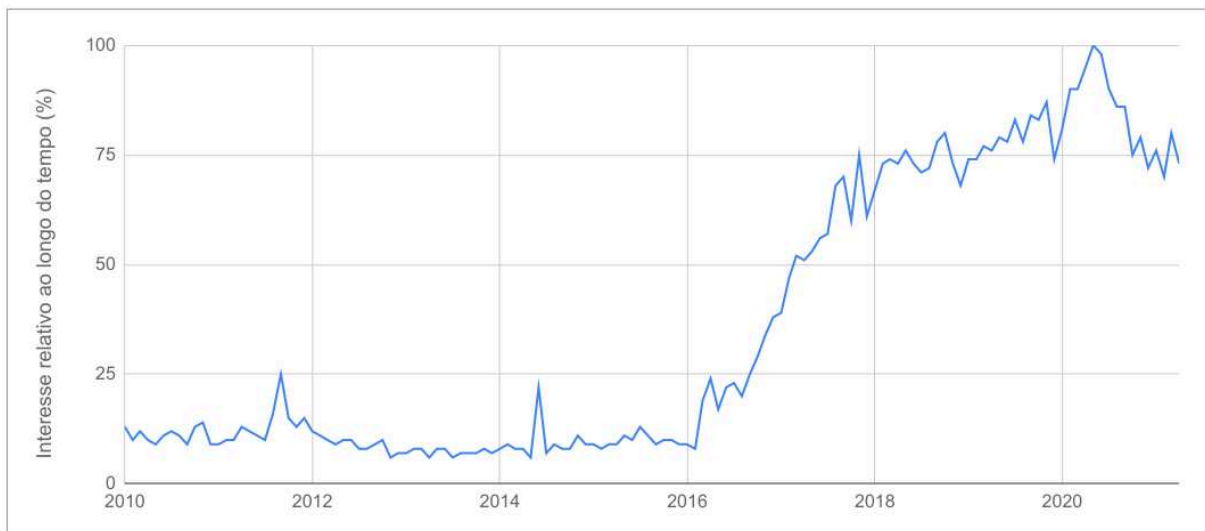


Figura 1 – Pesquisa pelo termo 'chatbot' no Google Trends. Os números representam o interesse de pesquisa relativo ao ponto mais alto no gráfico.

Apesar dos Chatbots existirem à décadas, sua popularidade aumentou exponencialmente nos últimos anos, mais precisamente a partir de 2016, conforme dados obtidos no através do Google Trends (Figura 1). Essa popularidade dos Chatbots vem em conjunto

¹ Linguagens que se desenvolveram naturalmente entre os seres humanos.

² Linguagens desenvolvidas sobre um conjunto de regras específicas, por exemplo, linguagens de programação.

com a maior quantidade de informações disponíveis na internet, o que acaba necessitando uma forma mais eficiente de acessá-la. Por exemplo, tradicionalmente, para que um usuário tenha uma dúvida sanada em um determinado sistema *web*, era necessário enviar um e-mail ao contato do site ou buscar a dúvida em um FAQ³. Com utilização de Chatbots, essas informações passaram a ser entregues de forma mais rápida, necessitando apenas a digitação do assunto ou pergunta em um campo de texto. Isso acaba facilitando o acesso e agilizando o processo de busca por informação.

Atualmente, uma das principais barreiras de desenvolvimento de Chatbots é o processamento da linguagem natural (PLN), ou seja, como extrair a informação de uma sentença fornecida pelo usuário de forma eficaz e coerente.

2.2 Processamento da Linguagem Natural (PLN)

Processamento da Linguagem Natural é um campo da ciência que estuda a interação entre computadores e a linguagem humana (ou linguagem natural). Mais precisamente, como os computadores podem extrair, processar e analisar dados de linguagem natural.

Um dos maiores desafios dos Chatbots é entender o sentido de uma sentença em linguagem natural, isso porque a linguagem humana é altamente ambígua e variável. Goldberg (2017) cita como exemplo as seguintes frases: "*Eu como pizza com amigos*" e "*Eu como pizza com azeitonas*". As frases são ambíguas, pois alterando apenas a palavra azeitona para amigos, o sentido completo da frase é alterado. Além disso, é altamente variável, pois a mesma mensagem passada pela frase "*Eu como pizza com amigos*" poderia ser passa com a frase "*Eu divido pizza com parceiros*". Outra característica importante é que a linguagem natural está em constante mudança e evolução (GOLDBERG, 2017).

As características das linguagens naturais citadas acima proporcionam um desafio para a construção de algoritmos e métodos computacionais capazes de interpretar o principal sentido de uma frase em meio à um diálogo. Contudo, tanto o desafio do entendimento quanto da produção de linguagem natural utilizando-se computadores pode ser enfrentado através de métodos e algoritmos de aprendizado de máquina que processam dados de linguagem (GOLDBERG, 2017).

A distância de Levenshtein é um método para calcular a diferença entre duas sequências de texto, esta técnica foi desenvolvida pelo matemático soviético Vladimir Levenshtein em 1965 (KASHEFI; SHARIFI; MINAIE, 2013; HOSSAIN et al., 2019). Este método é aplicado em diversas áreas onde existe a necessidade de comparação entre sequências de caracteres, sendo utilizado desde a comparação de sequências de DNA até corretores ortográficos (SANDERS; CHIN, 2009; HOSSAIN et al., 2019). No método

³ FAQ (em inglês *Frequently Asked Questions*) é uma seção de um sistema responsável por fornecer respostas para as perguntas mais frequentes.

de Levenshtein, quanto maior a diferença entre as sequências de caracteres comparados, maior será a distância de Levenshtein, logo, para duas palavras idênticas, a distância de Levenshtein é zero. A similaridade é calculada com base no número mínimo de operações requeridas para transformar uma palavra em outra, sendo que as operações disponíveis são a inserção, deleção e substituição de caracteres. Devido a capacidade de calcular a similaridade entre palavras, a distância de Levenshtein acaba sendo uma técnica muito utilizada por aplicações do tipo Chatbot (KOWSHER et al., 2019; RUF et al., 2020; MASLOWSKI; LAGARDE; CLAVEL, 2017).

Existem diversas bibliotecas para processamento de linguagem natural, como *spaCy* (HONNIBAL; MONTANI, 2021), *NLTK* (em inglês, *Natural Language Toolkit*) (BIRD EDWARD LOPER, 2021), *TextBlob* (LORIA, 2021), *Flair* (AKBIK; BLYTHE; VOLLGRAF, 2018), entre outras. A biblioteca *spaCy*, destaca-se por ser uma biblioteca de código aberto escrita em Python, que suporta mais de 64 linguagens, incluindo a língua portuguesa. O processamento é realizado através da quebra de textos em unidades menores, um processo chamado de *tokenização*. Como parte do processamento, a biblioteca também cria rótulos para cada unidade, denominando as unidades como substantivos, verbos, adjetivos, advérbios, etc, além de criar relações de dependências através da categorização de sujeitos e objetos. Além disso, a biblioteca também é capaz de deflexionar palavras para determinar o seu lema, por exemplo, o lema das palavras "visitarmos", "visitamos", "visitou" é "visitar". Por fim, a biblioteca também é capaz de verificar similaridade entre textos e classificá-los com rótulos e categorias. Essas funcionalidades da biblioteca *spaCy* devem-se principalmente à diversos algoritmos de aprendizado de máquina em sua composição. Portanto, devido às essas características, esta biblioteca é muito utilizada na construção de Chatbot (FINCH et al., 2020; MOPPEL, 2018).

2.3 Aprendizado de Máquina

O aprendizado de máquina (em inglês, *machine learning*) é o mecanismo pelo qual busca-se com que uma máquina tenha a capacidade de aprender sem programá-la explicitamente para isso (SINGH, 2018). Um exemplo simples para o entendimento de um sistema com aprendizado de máquina pode ser mostrado da seguinte forma. Treina-se uma máquina à diferenciar gatos de pássaros, mostrando diversas fotos diferentes de cada animal. Após o treinamento, ao apresentar uma foto de um dos animais, foto que até então não foi apresentada no treinamento, espera-se que a máquina saiba reconhecer se o animal é um gato ou um pássaro. O desafio que a máquina possui é ser capaz de aprender padrões ou abstrair características suficientes, durante o treinamento, para que ela seja capaz de prever algum padrão de forma correta (SINGH, 2018). No contexto de Chatbot, o aprendizado de máquina se aplica quando o sistema é treinado com conjuntos de perguntas e respostas e posteriormente testado. Após o treinamento, o Chatbot deve

ser capaz de responder não somente perguntas exatamente idênticas às treinadas, mas também perguntas similares que apresentam o mesmo sentido.

Atualmente existem diversas ferramentas de código aberto desenvolvidas para atuar como mecanismos (ou *engines*, em inglês) para sistemas de diálogo. A biblioteca Python ChatterBot (COX, 2021) é uma coleção de recursos que utilizam aprendizado de máquina e servem para esse propósito, sendo elegível para o desenvolvimento do presente trabalho.

3 Trabalhos Relacionados

Os Chatbots têm se tornado os canais de comunicação mais populares para as empresas, uma vez que os clientes interagem com a companhia de uma forma mais informal e em tempo real (KACZOROWSKA-SPYCHALSKA, 2019). Segundo o serviço de pesquisa *Business Insider Intelligence*, em uma pesquisa com mais de 800 representantes de empresas em diferentes continentes, 80% das instituições já utilizavam ou planejavam utilizar Chatbots até 2020 (INTELLIGENCE, 2016). Esse relatório mostrou a recente adesão e pretensão das instituições em utilizar a tecnologia de automação para facilitar o acesso à informação. Além do grande número de áreas em que os Chatbots podem ser aplicados, existe também a diversidade de bibliotecas e linguagens de programação em que os sistemas de conversação são desenvolvidos. Isso acaba por tornar o ecossistema de Chatbots muito amplo e de difícil comparação e/ou classificação.

Uma competição anual sobre inteligência artificial, chamada Loebner Prize (MAULDIN, 1994), premia programas de computadores com inteligência artificial que possuem maior semelhança com o ser humano. A competição utiliza o Teste de Turing¹ como forma de avaliação. Dentre os principais ganhadores, pode-se destacar dois Chatbots: Rose² e Mitsuku³. Ambos programas interagem utilizando a linguagem natural. Mitsuku, desenvolvida por Steve Worswick, vencedora dos últimos 4 anos (2016 a 2019), foi criada com a utilização da tecnologia AIML⁴. Já Rose, criada por Bruce Wilcox, vencedora nos anos 2014 e 2015, foi desenvolvida utilizando o mecanismo Chatscript⁵. Ao contrário do Chatbot proposto neste trabalho, Mitsuku e Rose não foram desenvolvidos para interagir sobre um conjunto de conhecimentos específicos, mas para conversações diversas sobre a maior quantidade de assuntos possíveis, que são constantemente incorporados em suas bases de dados sob uma grande quantidade de regras de conversação, além do idioma destes Chatbots serem em inglês.

No Brasil, os Chatbots são encontrados em diversas áreas, na maioria dos casos, como ferramentas de interação com usuários em plataformas de comércio eletrônico, com o objetivo de auxiliar em dúvidas de consumidores. No entanto, também são encontrados Chatbots com funções mais específicas, como procurar uma vaga de emprego, responder

¹ O Teste de Turing foi introduzido por Alan Turing em seu artigo na década de 50 (TURING; HAUGELAND, 1950), o qual procura avaliar a capacidade de um computador em simular o comportamento humano, onde o interrogador não percebe a diferença entre um ser humano e uma máquina através de uma comunicação não presencial (FRENCH, 2000).

² <http://brilligunderstanding.com>

³ <https://www.pandorabots.com/mitsuku>

⁴ Linguagem baseada em XML desenvolvida para criar diálogos semelhante a linguagem natural.

⁵ Combinação de Linguagem Natural com um sistema de gerenciamento de diálogo escrito na linguagem C++, desenvolvido por Bruce Wilcox para criação de Chatbots.

dúvidas sobre previdência privada ou auxiliar sobre a utilização de cartões de crédito (Tabela 1).

Tabela 1 – Comparação entre Chatbots de diferentes cenários.

Chatbot	Criador	Base de conhecimento	Tecnologia	Plataforma	Cenário
Rose ^a	Bruce Wilcox	Privado	Chatscript	Navegador	Demonstração
Mitsuku ^b	Steve Worswick	Parcialmente privado	AIML	Navegador, Facebook, Messenger, Telegram	Demonstração
Mr. Joboto ^c	Cavylabs	Privado	Não informado	Facebook, Messenger	Empresa de recrutamento
Otto ^d	ViaCognitiva	Privado	IBM Watson	Navegador	Previdência privada
VisaBot ^e	Smarters	Privado	Não informado	Facebook, Messenger	Cartões de crédito

^a <http://brilligunderstanding.com/rostedemo.html>

^b <https://www.pandorabots.com/mitsuku>

^c <http://www.mrjoboto.com>

^d <https://otto-weber.mybluemix.net>

^e <https://www.messenger.com/t/VisaBR>

Mr. Joboto é um exemplo de Chatbot direcionado à serviços oferecidos por uma instituição em sua área de negócio. Trata-se de um programa desenvolvido no Brasil em que o usuário interage com o Chatbot informando alguns dados sobre o perfil e localização. O robô realiza uma pesquisa sobre oportunidades de trabalho associadas ao perfil do usuário. No entanto, esse Chatbot não possui uma interface web própria, utilizando o *Messenger* do *Facebook* como meio de interação com o usuário. Da mesma forma, o Visabot é um canal de comunicação desenvolvido para tirar dúvidas sobre os cartões da empresa Visa, atuando como um assistente pessoal. O programa também interage com o usuário através da plataforma *Messenger* para tirar dúvidas de clientes, ou possíveis clientes, sobre as tecnologias da empresa, ofertas, benefícios, entre outras informações.

Otto, por sua vez, é um Chatbot que possui uma interação direta com o usuário através do navegador de internet. Esse Chatbot foi desenvolvido com o objetivo de interagir com o consumidor, nele é possível tirar dúvidas sobre temas relacionados à previdência privada. O sistema foi desenvolvido através da plataforma IBM Watson, que consiste em uma plataforma de serviços cognitivos da IBM capaz de interagir através de perguntas e respostas em linguagem natural. No entanto, trata-se de uma tecnologia paga e de código fechado.

Os Chatbots citados acima são, na maioria dos casos, programas direcionados para soluções específicas, seja para uma área em especial, serviço ou simplesmente para conversação geral com o usuário. Além disso, muitos deles utilizam plataformas como *Telegram* ou *Facebook*, tornando o Chatbot dependente desses sistemas para seu funcionamento.

Portanto, nenhum dos programas mencionados poderia ser diretamente aplicado para a solução proposta no trabalho, apesar da possível utilização de bibliotecas ou linguagens comuns. O presente trabalho necessitará cadastrar uma base de conhecimento própria, desenvolver um painel administrativo e uma interface de *chat*.

Recentemente, Xie et al. (2019) utilizaram a biblioteca Python *ChatterBot* para construir um programa com o objetivo de fornecer informações sobre criptomoedas. O material utilizado para o treinamento do Chatbot foi dados obtidos através do Quora, um website de perguntas e resposta. Para coletar o máximo de informações possível, foi utilizado um *crawler*⁶ desenvolvido em Python que buscava por perguntas e respostas associadas ao tema de criptomoedas. Os pesquisadores utilizaram um total de mais de dez mil perguntas e respostas para o treinamento sem supervisão e o valor de confiança determinado na configuração da biblioteca foi definido manualmente entre 0,7 e 0,8. Esse valor define a precisão da resposta, ou seja, se o grau de confiança for muito baixo, é possível que uma pergunta mal compreendida possa retornar uma resposta incorreta. No entanto, o artigo não mostra de forma explícita como esse valor foi encontrado metodologicamente, mencionando que após alguns testes o valor determinado foi o que apresentou melhor cenário. Por fim, os autores consideraram que os testes mostraram-se promissores em responder questões de usuários relacionadas à criptomoedas. Esse resultado mostra que o uso da biblioteca *ChatterBot* é promissor em relação à construção de sistemas de Chatbot.

Semelhante ao Chatbot de Xie et al. (2019), o presente trabalho também utiliza a biblioteca Python *ChatterBot* para o desenvolvimento do sistema, no entanto, além do conteúdo do diálogo ser diferenciado em relação à criptomoedas, as combinações de perguntas e respostas precisam ser inseridas manualmente através de um painel administrativo, ao contrário do sistema de Xie, onde a base de conhecimento foi extraída de um site de perguntas e respostas e introduzidas diretamente na biblioteca. Uma vez que o presente trabalho trata de assuntos específicos do curso de graduação de Sistemas de Informação, necessitando uma interface administrativa amigável para cadastro de novos conhecimentos, treinamento e supervisionamento das interação entre usuário e Chatbot.

⁶ Crawlers são programas de computador que sistematicamente navegam pela internet de forma automatizada e com propósitos pré-definidos. Realizando o download de páginas da internet, os crawlers podem extrair conteúdos que tangem seu objetivo.

4 Desenvolvimento

4.1 Especificação dos Requisitos de Software

Nesta seção são apresentados os requisitos de software que definem as características e condições sob as quais o sistema deve funcionar. Os requisitos foram levantados por meio de prototipação e reuniões com o orientador do trabalho, onde foram identificados os requisitos para o funcionamento do sistema.

4.1.1 Requisitos Funcionais

Os requisitos funcionais descritos abaixo definem as necessidades e exigências em termos de funcionalidade, ou seja, o que o sistema Chatbot se propõe a realizar.

- RF001 - Perguntar ao Chatbot.
O sistema deve fornecer uma interface de bate-papo onde o usuário possa realizar perguntas em formato texto.
- RF002 - Responder a pergunta do usuário.
O sistema deve retornar uma resposta em formato texto ao bate-papo para toda a sentença enviada pelo usuário.
- RF003 - Solicitar *feedback* da resposta do Chatbot.
A interface de bate-papo deve solicitar um *feedback* do usuário sobre a resposta enviada pelo Chatbot.
- RF004 - Avaliar a interação com o Chatbot.
A interface de bate-papo deve fornecer a possibilidade de uma avaliação final da interação com o Chatbot.
- RF005 - Armazenar o bate-papo.
O sistema deve armazenar o bate-papo entre o usuário e o Chatbot para posterior supervisão.
- RF006 - Acessar a área administrativa.
O sistema deve possuir uma área administrativa com acesso por meio de usuário e senha.
- RF007 - Cadastrar perguntas e respostas.
O sistema deve fornecer ao administrador a possibilidade de cadastrar perguntas e respostas para aumentar a base de conhecimento do Chatbot.

- RF008 - Treinar o Chatbot.
O sistema deve fornecer ao administrador a possibilidade de treinamento do Chatbot através das perguntas e respostas previamente cadastradas.
- RF009 - Visualizar interações por usuário.
O sistema deve fornecer ao administrador a possibilidade de observar cada pergunta e resposta efetuada no bate-papo.
- RF010 - Visualizar *feedback* das respostas.
O sistema deve fornecer ao administrador a possibilidade de observar o *feedback* do usuário para cada resposta fornecida à ele.
- RF011 - Visualizar avaliação final dos usuários.
O sistema deve fornecer ao administrador a possibilidade de observar a avaliação final de uma conversa.

4.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem as condições que o sistema funcionará dentro do escopo definido, ou seja, são os requisitos que não definem as funcionalidades do sistema.

- RNF001 - Interação do usuário com o Chatbot.
A interface de bate-papo do Chatbot deve ser acessada via navegador de internet, compatível com versões atuais de Google Chrome e Mozilla Firefox.
- RNF002 - Acesso ao Chatbot.
A interface de bate-papo do Chatbot deve ser multiplataforma, sendo acessada da mesma forma independente do sistema operacional.
- RNF003 - Execução dos módulos do sistema.
Os módulos do sistema executados no lado do servidor devem utilizar um sistema operacional do tipo Linux.
- RNF004 - Infraestrutura dos módulos do sistema.
Os módulos do sistema devem executar em forma de contêineres Docker.
- RNF005 - Biblioteca de Chatbot.
A biblioteca para o mecanismo de Chatbot deve ser do tipo *open source*.
- RNF006 - Design responsivo na interface gráfica do bate-papo.
A interface gráfica deve ser funcional independente do dispositivo utilizado, navegador *Desktop* ou *Smartphone*.

- RNF007 - Segurança na API.

O sistema deve implementar um mecanismo de segurança para impedir a injeção de códigos SQL's nos métodos da API.

4.1.3 Regras de Negócio

Nesta seção são apresentadas as regras de negócio que definem como o sistema atenderá às necessidades definidas. Estas regras compreendem as restrições e premissas da operação do *software*, que garantem que o sistema funcione da forma proposta.

- RN001 - Interação com Chatbot.

O usuário poderá somente interagir com o Chatbot por meio de um formulário para inserção de texto através de uma página acessada via navegador de internet, sem necessidade de cadastro prévio.

Dependências: RF001

- RN002 - Respostas do Chatbot.

O sistema deve responder a toda mensagem enviada pelo usuário. Nos casos em que o sistema não conseguiu identificar uma resposta adequada para uma pergunta, deverá ser retornada uma resposta padrão ao usuário informando que não houve entendimento ou não sabe a resposta para a solicitação.

Dependências: RF002

- RN003 - Avaliação individual de uma resposta.

O usuário deve obrigatoriamente fornecer um *feedback* (positivo ou negativo) para cada resposta fornecida pelo Chatbot. Enquanto o usuário não fornecer o *feedback*, ele deve ficar impossibilitado de realizar a próxima pergunta.

Dependências: RF001, RF003

- RN004 - Avaliação final de uma interação.

A interface deve fornecer a possibilidade do usuário realizar uma avaliação geral da interação com o Chatbot. Essa avaliação deve ser feita através de uma seleção de valor entre uma faixa de 1 e 5, onde 1 é muito insatisfeito e 5 muito satisfeito.

Dependências: RF004

- RN005 - Persistência dos dados dos usuários.

O sistema deve armazenar toda pergunta e resposta para cada interação entre um usuário e o Chatbot. O armazenamento deve utilizar um identificador único para cada sessão aberta no navegador, possibilitando assim a identificar todas as perguntas e respostas de uma interação completa de uma sessão do usuário, juntamente com o registro da data e horário.

Dependências: RF005

- RN006 - Privacidade dos dados dos usuário.
O usuário não necessita fornecer nenhum dado pessoal para ter a possibilidade de interagir com o Chatbot. Uma vez que o usuário abre uma sessão no navegador da interface de bate-papo, um identificador alfanumérico único é gerado e armazenado junto com as perguntas e respostas para associá-las à uma sessão única de interação.
Dependências: RF005
- RN007 - Acesso administrativo.
O acesso à área administrativa deve ser realizado com um usuário e senha previamente cadastrados na instalação do sistema. Com o usuário e senha fornecidos é aberta uma sessão temporária no navegador para permitir o acesso à área de administração do sistema.
Dependências: RF006
- RN008 - Conhecimento do Chatbot.
A base de conhecimento do Chatbot deve ser baseada em perguntas e respostas cadastradas na área administrativa do sistema. O sistema deve permitir o cadastro de variações de perguntas para uma determinada resposta.
Dependências: RF007
- RN009 - Treinamento do Chatbot.
O sistema não deve treinar automaticamente o Chatbot após a inserção de novas perguntas e respostas, esse evento deve ser disparado manualmente através de um botão na interface da área administrativa do sistema.
Dependências: RF007, RF008
- RN010 - Visualização de uma interação completa.
Ao visualizar as perguntas e repostas das interações entre o usuário e o Chatbot, o administrador pode agrupar as perguntas e respostas através do identificador único de sessão, fornecendo uma visualização de toda a interação de um único usuário.
Dependências: RF009
- RN011 - Visualização das perguntas e respostas.
O texto da pergunta e da resposta, a data e hora do registro e, o *feedback* positivo ou negativo são fornecidos ao administrador através de uma seção da área administrativa. O administrador poderá realizar buscas nessa listagem através de um campo de texto e também realizar a ordenação por coluna dessa tabela.
Dependências: RF009, RF010
- RN012 - Visualização das avaliações.
Ao visualizar as avaliações finais dos usuários, o administrador terá acesso à data e hora da avaliação e ao identificador único da sessão. O identificador único possibilitará

a visualização através de um filtro das perguntas e respostas dessa sessão avaliada. O administrador poderá realizar buscas nessa listagem através de um campo de texto e também realizar a ordenação por coluna dessa tabela.

Dependências: RF009, RF011

4.2 Arquitetura do Sistema

O sistema Chatbot foi dividido em três serviços (ou módulos) denominados: Chatbot API, Chatbot Admin e Chatbot Frontend. Além dos serviços do sistema, a arquitetura também é composta por um banco de dados relacional, o qual persistirá todos os dados pertinentes ao sistema, incluindo a persistência requerida pela biblioteca Python ChatterBot. A Figura 2 mostra uma visão geral de cada módulo e uma descrição de suas principais atribuições. As seções abaixo descrevem cada um dos serviços que compõe o sistema, juntamente com as tecnologias envolvidas.

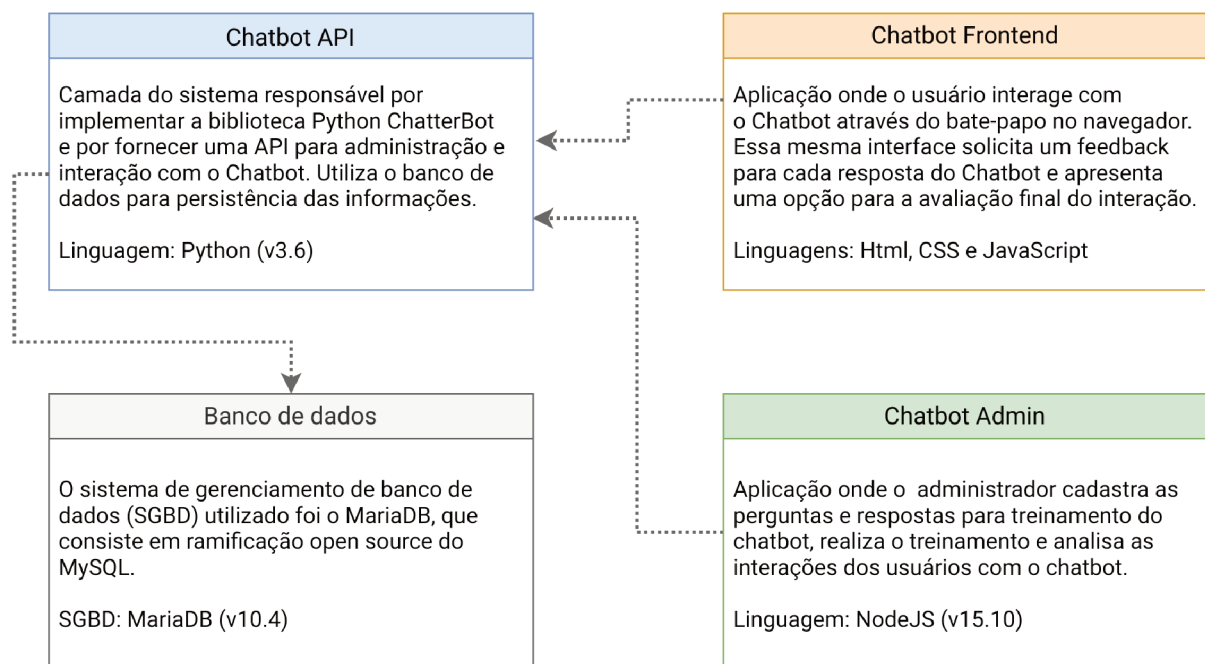


Figura 2 – Serviços que compõe o sistema proposto no trabalho.

4.2.1 Serviço Chatbot API

A API do Chatbot foi desenvolvida utilizando a linguagem Python. A escolha da linguagem foi realizada devido à existência de uma biblioteca de código aberto, denominada ChatterBot. Essa biblioteca foi desenvolvida para construção de respostas automatizadas em linguagem natural (COX, 2021), para isso, a ferramenta utiliza uma seleção de

algoritmos de aprendizado de máquina para produzir diferentes tipos de respostas. A biblioteca ChatterBot também permite o treinamento do sistema com diálogos montados na forma de pergunta e resposta. Além de permitir a utilização de diversos adaptadores de armazenamento, fornecendo assim uma flexibilidade para a escolha do banco de dados em que os diálogos serão persistidos.

Com a definição do mecanismo escrito em Python, optou-se por manter a mesma linguagem de programação para a construção da API, utilizando o *framework Flask* (RONACHER, 2021). Esse *framework* possui ferramentas que auxiliam no desenvolvimento de uma API com estilo arquitetural REST¹.

4.2.2 Serviço Chatbot Frontend

O módulo Chatbot Frontend consiste na interface responsável pela interação do usuário com o Chatbot. A interação é realizada através de uma página acessada pelo navegador de internet, onde é fornecido um formulário para interação com o *chatbot* via texto.

As tecnologias envolvidas no desenvolvimento desse serviço consistem em HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript. Optou-se por essas tecnologias por serem interpretadas no navegador do usuário, sem a necessidade de um processamento no servidor, além do fato de serem tecnologias simples e amplamente utilizadas para criação desse tipo de interface. Para auxiliar no desenvolvimento optou-se também pela utilização de uma biblioteca de funções JavaScript chamada jQuery.

4.2.3 Serviço Chatbot Admin

Para o desenvolvimento do serviço de administração optou-se pela linguagem Node.js (DAHL, 2021). Uma vez que esse sistema necessita comunicar-se com o serviço Chatbot API e também fornecer uma interface para o usuário administrador realizar os procedimentos com boa usabilidade, a linguagem Node.js apresenta-se como uma boa alternativa código aberto, possuindo bibliotecas que facilitam o desenvolvimento de sistemas que necessitam de um processamento no servidor e também um fornecimento de arquivos estáticos (interface de usuário).

4.2.4 Persistência de dados

O armazenamento dos dados do *chatbot* é realizado com a utilização do Sistema de Gerenciamento de Banco de Dados (SGBD) MariaDB na versão 10.4.17 (WIDENIUS,

¹ REST (em inglês *Representational State Transfer*) é um estilo de arquitetura de *software* para sistemas *web*, que fornece um conjunto de restrições arquitetônicas para enfatizar a escalabilidade das interações de componentes, generalizar interfaces e tornar componentes independentes (ROY, 2000).

2021). O banco de dados MariaDB é uma ramificação do projeto MySQL após sua aquisição pela Oracle Corporation, dessa forma, o projeto MariaDB permanece gratuito e de código aberto. Trata-se de um banco de dados do tipo relacional, que possui uma fácil integração com a linguagem Python, sendo também compatível com a biblioteca Python ChatterBot.

4.2.5 Orquestração do sistema via Docker Compose

Com o objetivo de manter os serviços do sistema com alta portabilidade, optou-se pela utilização de containerização com Docker (HYKES, 2021). Desta forma, os serviços podem ser executados em qualquer sistema que possua o docker instalado, executando diretamente o contêiner ou através de orquestradores de contêineres. Além da portabilidade, o containerização garante uma boa performance e escalabilidade dos serviços, além de manter o ambiente da aplicação isolado e efêmero, garantindo propriedades de segurança.

Cada serviço do sistema é executado em um contêiner Docker (Figura 3). Uma vez que as aplicações se comunicam via REST, cada serviço (módulo) do sistema pode ser executado em um local (ou máquina) diferente. No entanto, o sistema também pode ser executado em uma única máquina utilizando-se um orquestrador de contêineres Docker, como o Docker Compose. O Apêndice A mostra o arquivo do tipo YAML² para orquestração dos contêineres de cada módulo.

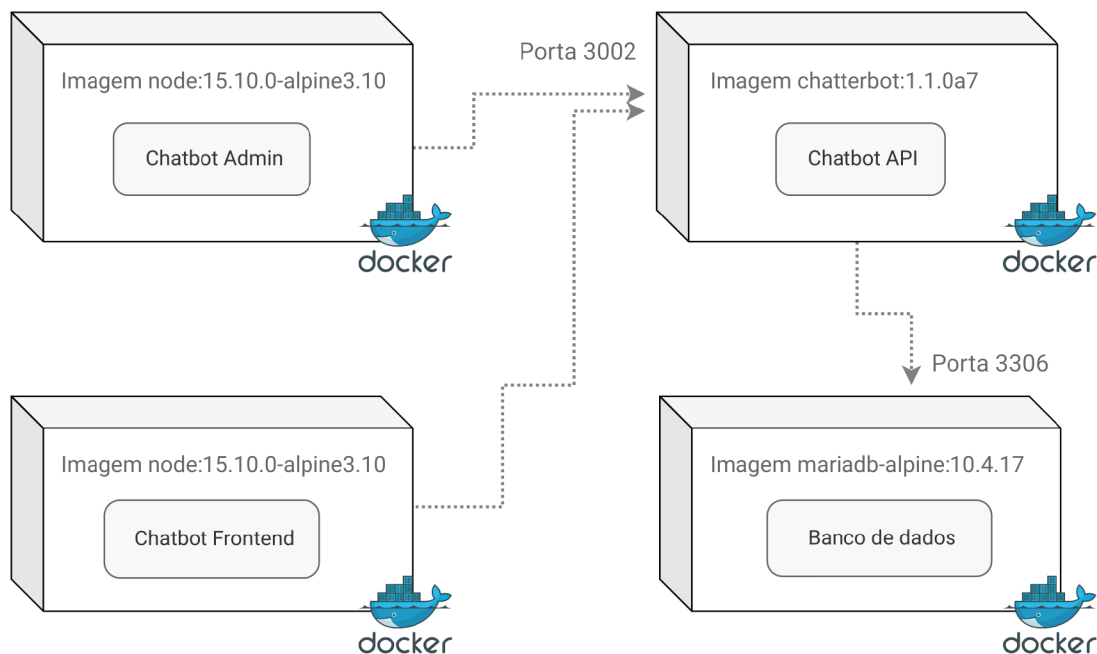


Figura 3 – Cada serviço do sistema é executado de forma independente e isolada através de contêineres que executam imagens Docker, específicas para cada serviço.

No módulo Chatbot Frontend, a aplicação é executada na porta 3000 utilizando a imagem Docker oficial do NodeJS na versão *15.10.0-alpine3.10*. Como variável de ambiente,

² YAML é um formato de serialização de dados.

destacam-se uma variável que determina o endereço HTTP da API e uma variável que determina se o bate-papo deve solicitar o *feedback* para cada resposta dever ser obrigatório antes de permitir uma próxima pergunta.

No módulo Chatbot Admin, o serviço é exposto na porta 3001 utilizando a mesma imagem Docker do módulo Chatbot Frontend. Neste módulo destacam-se as variáveis de ambiente que definem o endereço HTTP da API e as variáveis utilizadas como chaves para manter a sessão criptografada entre o navegador do usuário administrador e o servidor.

No módulo Chatbot API, a aplicação Python é executada utilizando-se a porta 3002 em uma imagem construída com base na imagem oficial do Python (*python:3.7.9-alpine3.13*) a qual foi nomeada *chatterbot:1.1.0a7*. O código da imagem criada pode ser observado no Apêndice B. Nesse *Dockerfile* estão descritas as bibliotecas e pacotes necessários para a execução correta do mecanismo de Chatbot (Python Chatterbot), do *framework Flask*, bibliotecas de criptografia e integração com banco de dados do tipo MySQL. As variáveis de ambiente da aplicação definem a conexão com o banco de dados, o usuário e senha de administrador, a chave de geração de *token* e o tempo de expiração dele.

Finalmente, o último serviço é executado na porta 3306 em uma imagem do SGBD MariaDB, versão 10.4.17. Esse serviço de banco de dados comunica-se apenas com o módulo Chatbot API.

4.3 Modelagem

4.3.1 Diagrama de Entidade Relacionamento

O Diagrama Entidade Relacionamento (DER) (Figura 4) representa visualmente as entidades do banco de dados de todo o sistema Chatbot. As tabelas *statement*, *tag* e *tag_association* são criadas automaticamente pela biblioteca Python ChatterBot. Já as demais tabelas foram modeladas a fim de alcançar as funcionalidades do sistema. A tabela *user* armazena os dados dos usuários administrativos, que consiste no nome do usuário, um *login* e uma senha criptografada. A tabela *training* armazena todos os pares de perguntas e respostas que são utilizados para treinar a biblioteca ChatterBot. A tabela *rating_response* armazena cada sentença de pergunta e resposta entre o usuário e o Chatbot, juntamente com o *feedback* para a resposta e um identificador único da sessão do usuário. Por fim, a tabela *rating_final* persiste a avaliação final realizada pelo usuário para sua interação com o Chatbot e o identificador único da sessão. O identificador de sessão é utilizado para relacionar toda a interação do usuário com a avaliação final da interação.

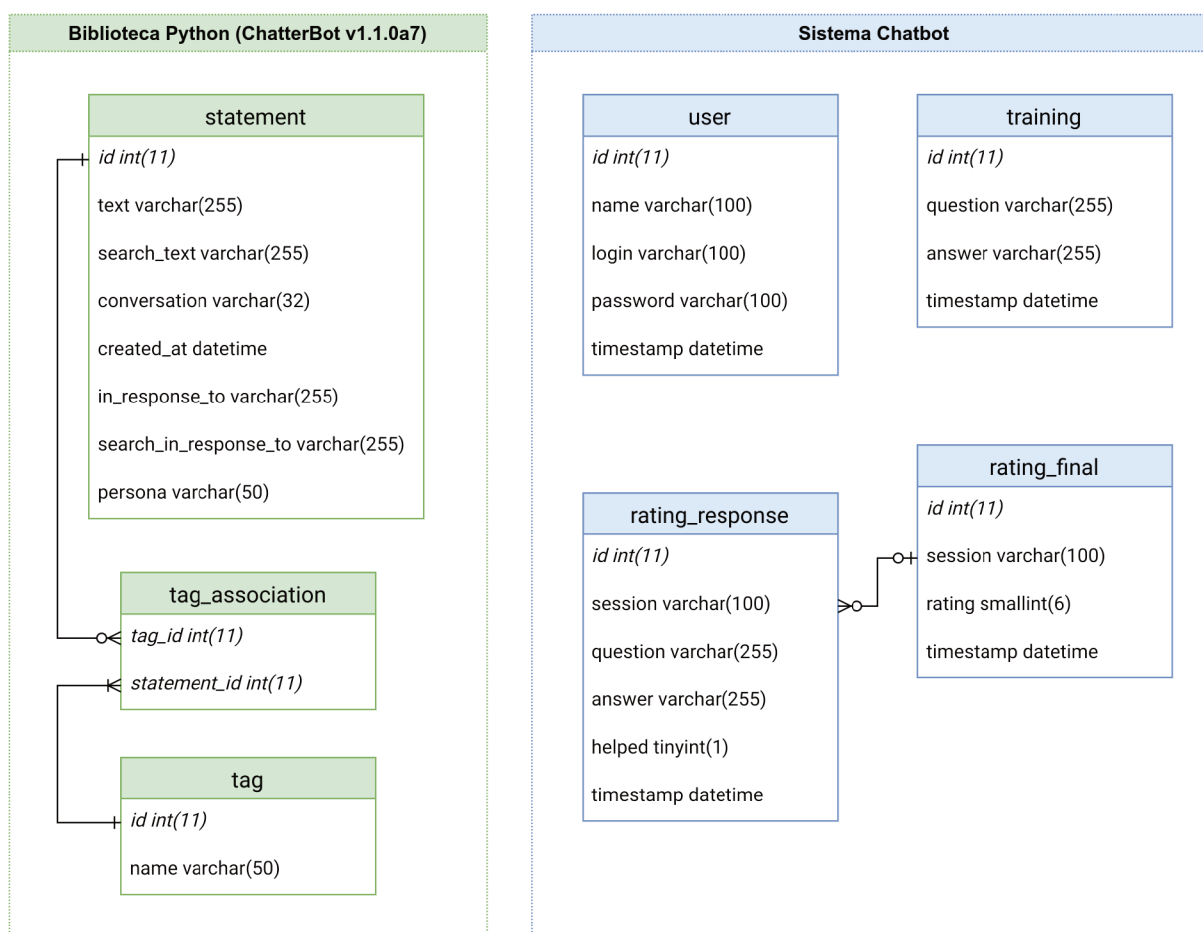


Figura 4 – Representação visual das entidades do banco de dados do sistema através do diagrama ER.

4.3.2 Diagrama de Atividades

A Figura 5 apresenta o diagrama de atividade da interação do usuário com o Chatbot, mostrando a sequência de ações que ocorrem nessa interação. Inicialmente o usuário acessa a página *web* do Chatbot, ao clicar no ícone referente à abertura do bate-papo, o usuário tem a possibilidade de escrever uma mensagem de texto no campo da caixa de bate-papo. Em seguida, a aplicação retorna uma resposta para o usuário através de uma mensagem na mesma caixa de bate-papo. Para que o usuário possa escrever outra mensagem, é obrigatória a avaliação da mensagem retornada pelo Chatbot, selecionando um ícone com valor positivo ou negativo. O usuário pode continuar a interagir com o Chatbot, avaliando cada resposta, ou pode selecionar a opção "Encerrar e Avaliar", a qual abrirá uma segmento da tela para a avaliação da interação através da escolha de um valor de 1 a 5.

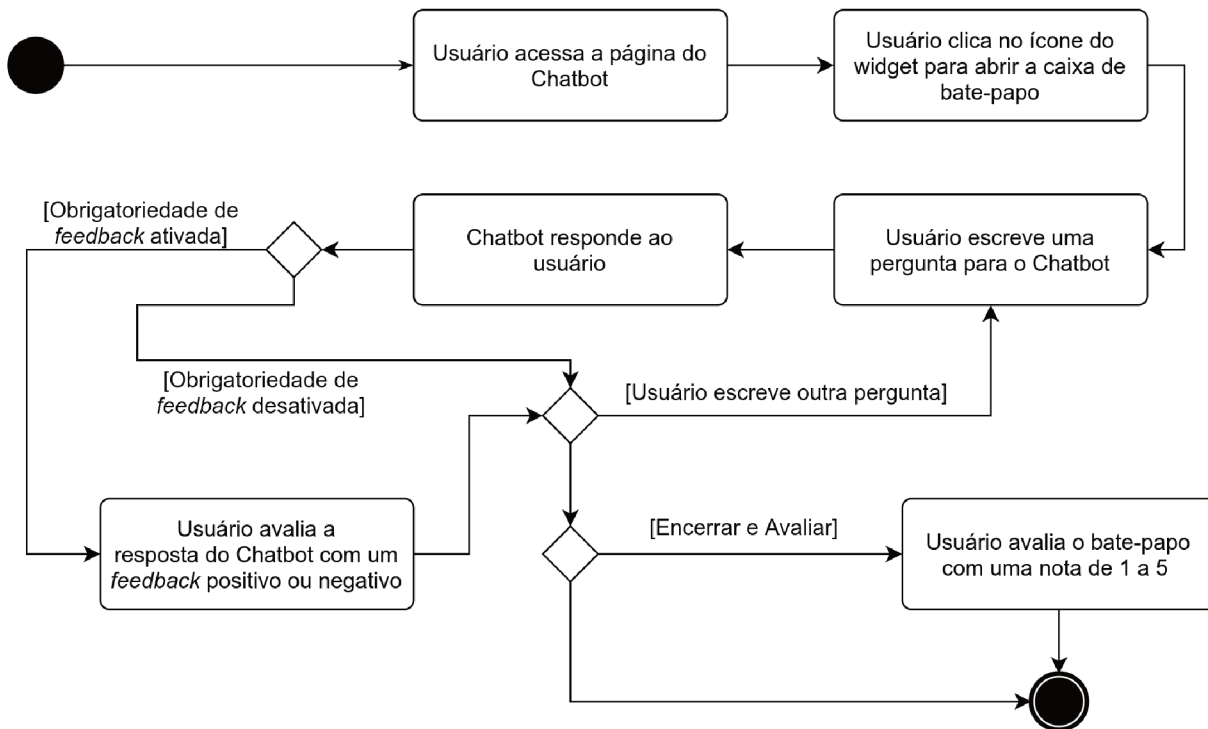


Figura 5 – Diagrama de atividade UML relacionado à interação do usuário com o Chatbot.

A Figura 6 apresenta o diagrama de atividades que representa a sequência de ações para realizar o aumento do conhecimento do Chatbot e seu treinamento. Para realizar essa tarefa é necessário acessar a página *web* da área administrativa, efetuar o *login* através do usuário e senha. Uma vez conectado à área administrativa, o administrador deve acessar o menu "Treinamento". Na tela de treinamento, o administrador pode observar os pares de pergunta e resposta previamente cadastrados e optar por treinar o Chatbot ou cadastrar novas informações. Ao optar por cadastrar novas informações, o administrador será direcionado à um formulário para cadastro de pergunta e resposta. Nessa etapa, o administrador pode inserir variações de perguntas para uma única resposta. Após o cadastro, o administrador voltará a tela de treinamento e poderá optar por treinar o Chatbot, incorporando assim as novas informações no banco de dados da biblioteca.

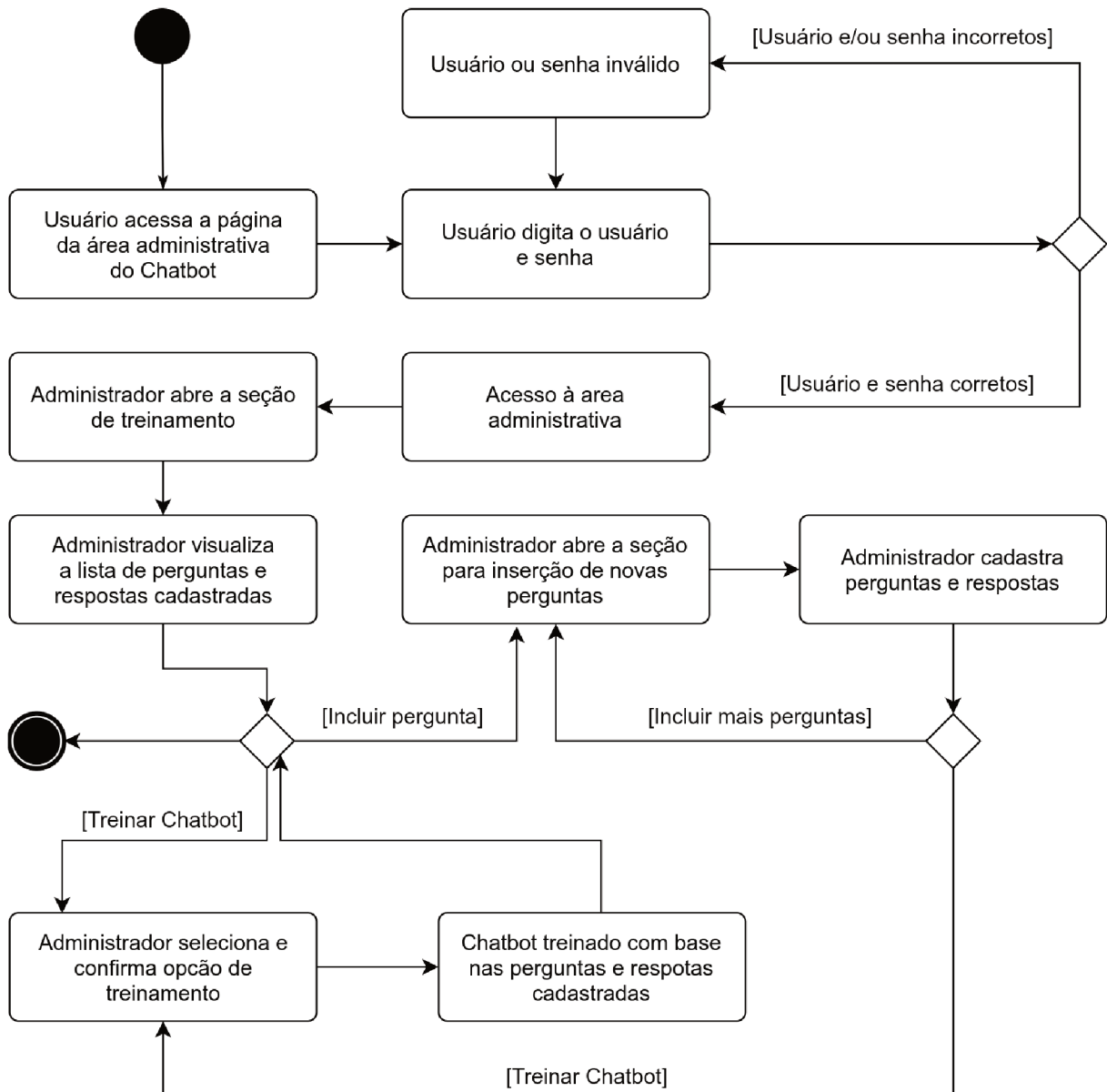


Figura 6 – Diagrama de atividade UML relacionado à ação de treinamento realizada pelo administrador.

4.4 Biblioteca Python ChatterBot

A biblioteca Python ChatterBot fornece ferramentas que auxiliam na geração automatizada de respostas. Esta biblioteca permite uma configuração a fim de customizar a aplicação para uma melhor experiência do usuário. Além de permitir diferentes formas de aprendizado, também fornece a opção para diferentes tipos de algoritmos de aprendizado de máquina, a fim de produzir diferentes tipos de respostas, na maioria dos casos, algoritmos de classificador Naïve Bayes. Ou seja, o sistema utiliza técnicas de aprendizado de máquina para selecionar e classificar respostas que possuem a correspondência mais próxima à

sentença de entrada, escolhendo então a melhor resposta como sentença de saída.

Outra característica importante da biblioteca Python ChatterBot é a sua arquitetura independente de idioma, dessa forma, permite que o Chatbot seja treinado com qualquer idioma. Esse aspecto é muito importante, uma vez que o objetivo do trabalho está direcionado para um Chatbot no idioma português.

Portanto, devido à biblioteca Python ChatterBot ser um software de código aberto, ser de fácil utilização, possuir vasta documentação, além das características citadas nos parágrafos anteriores, optou-se por utilizar esta biblioteca no presente trabalho como mecanismo para automatizar o sistema do Chatbot.

4.4.1 Funcionamento da biblioteca Python ChatterBot

O treinamento executado na biblioteca Python ChatterBot pode ser realizado através da inserção de pares de perguntas e respostas. O mecanismo utilizado está baseado na utilização de outra biblioteca de PLN, spaCy (descrita na seção 2.2). Desta forma através da spaCY, a biblioteca Python ChatterBot é capaz de armazenar as sentenças com extração do lema das palavras fornecidas, geração de *tokens* e remoção de *stop words*, que são as palavras que tem baixa influência no sentido da frase, como: "o", "a", "é", entre outras.

Após o treinamento, a biblioteca está preparada para receber sentenças por meio de entradas em forma de texto. Esse texto então é processado através de adaptadores lógicos, no fim, a melhor resposta é entregue na saída da aplicação (Figura 7).

Os adaptadores lógicos determinam a lógica de como o Chatbot escolhe qual é a melhor resposta para a sentença de entrada. Adaptadores lógicos podem apresentar funções diferentes, por exemplo, identificar se a sentença é um problema matemático, uma conversa trivial, uma pergunta plausível, entre outros. Sendo cada adaptador lógico responsável por um domínio, a sentença de entrada irá passar por todos os adaptadores, onde, cada um deles fornece um valor de confiança junto com a melhor resposta para a sentença.

A biblioteca ChatterBot permite a criação de adaptadores lógicos, no entanto, ela fornece uma série de adaptadores prontos. Por exemplo, o *Time Logic Adapter* serve para identificar perguntas relacionadas a hora atual. Já o *Mathematical Evaluation Adapter* identifica sentenças em que é necessário realizar algum cálculo matemático. A seguir tem-se um exemplo de entrada e saída da biblioteca utilizando-se os adaptadores lógicos de tempo e cálculo matemático, respectivamente:

Usuário: Que horas são?

Chatbot: "Agora são exatamente 19:57."

Usuário: Quanto é dois mais dois?

Chatbot: $(2 + 2) = 4$

O adaptador lógico *Best Match Adapter* busca por uma resposta em que a pergunta possua a correspondência mais próxima à pergunta de entrada, utilizando uma função que compara sentenças. A sentença com maior índice de similaridade é utilizada. Em casos em que o adaptador encontra sentenças com o mesmo índice de similaridade, a escolha pela resposta ocorre de forma randômica. Este foi o adaptador lógico escolhido para o presente trabalho.

O método de comparação de sentenças também pode ser criado ou alterado dentre diferentes métodos disponíveis na biblioteca. De forma padrão, a biblioteca utiliza a distância de *Levenshtein*, descrita na seção 2.2, para comparar as sentenças. Com base nessa distância, o Chatbot calcula o grau de similaridade entre a entrada e os registros da base de dados do Chatbot.

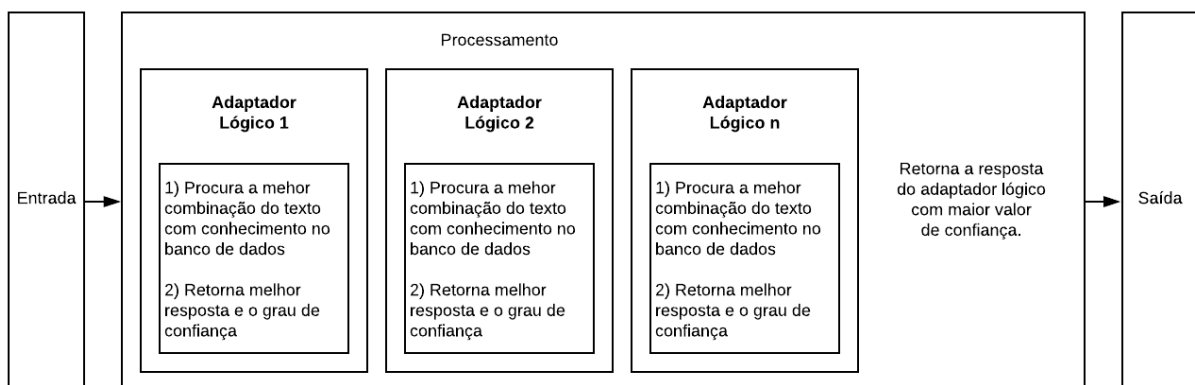


Figura 7 – Biblioteca Python Chatterbot.

Por fim, o processamento da entrada é finalizado com a entrega da resposta com o maior valor de confiança encontrado pelos adaptadores lógicos. O valor de confiança varia entre 0 e 1, sendo que seu limite deve ser imposto manualmente. Dessa forma, se a melhor resposta obtiver um valor menor do que o limite imposto, ela será ignorada e o Chatbot apresentará uma mensagem padrão configurada.

Valores de confiança muito baixos podem fornecer respostas incorretas. Por outro lado, um valor de confiança muito próximo a 1 torna o Chatbot muito rigoroso com as combinações, não aceitando erros de ortografia ou palavras semelhantes. Assim, diferentes valores de confiança devem ser testados para obter a melhor combinação de perguntas e respostas.

No exemplo da Tabela 2, a biblioteca foi treinada com a pergunta "*Qual é a localização do Departamento de Informática e de Estatística (INE)?*" e com a resposta "*O INE está localizado no Centro Tecnológico (CTC) da UFSC.*". Foram testadas algumas variações de perguntas, descritas na tabela. Para pergunta idêntica, o valor de confiança atingiu o resultado de $1,0$, para as demais perguntas, o resultado da confiança foi decrescendo conforme diminuía a similaridade das sentenças. Desta forma, para um grau de confiança hipoteticamente definido no valor de $0,5$, a biblioteca responderia com a resposta treinada para as quatro primeiras perguntas (valores de confiança $1,0$, $0,82$, $0,73$ e $0,55$), ignorando a última com um valor de confiança de $0,44$, retornando para esta uma mensagem padrão personalizada.

Tabela 2 – Exemplo de treinamento e teste da biblioteca Python *Chatterbot*.

<p>Treinamento:</p> <p>Pergunta: Qual é a localização do Departamento de Informática e de Estatística (INE)?</p> <p>Resposta: O INE está localizado no Centro Tecnológico (CTC) da UFSC.</p>
<p>Teste:</p> <p>Pergunta: Qual é a localização do Departamento de Informática e de Estatística (INE)? [<i>Confiança = 1.0</i>]</p> <p>Pergunta: Qual é a localização do Departamento de Informática? [<i>Confiança = 0.82</i>]</p> <p>Pergunta: Localização do Departamento de Informática? [<i>Confiança = 0.73</i>]</p> <p>Pergunta: Qual é o significado da sigla INE? [<i>Confiança = 0.54</i>]</p> <p>Pergunta: Qual é a localização? [<i>Confiança = 0.44</i>]</p>

4.5 Implementação

Cada módulo do sistema atua como um serviço *web* independente, ou seja, eles podem ser executados em diferentes locais. Os módulos utilizam o padrão arquitetural REST (*Representational State Transfer*), dessa forma comunicam-se entre eles através dos métodos HTTP. As solicitações e respostas realizadas entre os módulos são realizadas utilizando-se o formato JSON (*JavaScript Object Notation*). Desta forma, cada serviço pertencente ao sistema foi desenvolvido separadamente. O código fonte de cada um dos serviços está disponível em um repositório público nos seguintes endereços:

- **Chatbot:** <https://codigos.ufsc.br/thiago.ps/chatbot>
- **Chatbot Api:** <https://codigos.ufsc.br/thiago.ps/chatbot-api>
- **Chatbot Admin:** <https://codigos.ufsc.br/thiago.ps/chatbot-admin>
- **Chatbot Frontend:** <https://codigos.ufsc.br/thiago.ps/chatbot-frontend>

4.5.1 Serviço Chatbot API

O módulo Chatbot API pode ser considerado o núcleo do sistema, uma vez que é o local onde a biblioteca Python ChatterBot é implementada. Ao entorno dessa biblioteca, foi desenvolvida uma API que utiliza um micro *framework* denominado Flask que interage com a biblioteca ChatterBot e realiza persistência dos dados no banco de dados.

A biblioteca Python ChatterBot utiliza três tabelas no banco de dados, essas tabelas são preenchidas quando é realizado um treinamento na biblioteca. O Treinamento consiste no envio de uma lista de pares de pergunta e resposta, onde a biblioteca processa esses dados e cadastra em suas tabelas. Para que fosse possível o cadastro de novas informações na biblioteca, foi criada uma tabela no banco de dados chamada *training*. Essa tabela armazena uma lista de pergunta e resposta, onde os registros são adicionados ou removidos através de chamadas para a API, ou pela interface administrativa, a qual interage com essa API. Portanto, o treinamento do Chatbot consiste na leitura dessa tabela *training* para processamento da biblioteca e inserção de registros nas tabelas relacionadas à biblioteca.

A maioria das chamadas à API necessitam de autenticação, a qual é realizada através de uma chamada à API passando um usuário e senha. Esse usuário e senha foram criados durante a execução do sistema, através de variáveis de ambiente do serviço. Ao realizar a chamada de autenticação, o usuário recebe um *token* que deve ser utilizado no cabeçalho das chamadas que necessitam de autenticação.

Uma vez que alguns métodos da API não necessitam de autenticação, questões de segurança tornam-se importantes. Para que não haja injeção de códigos SQL maliciosos, utilizou-se biblioteca *SQLAlchemy* para mapeamento objeto-relacional (ORM, do inglês: *Object-relational mapping*) SQL, que impede a inserção de caracteres passíveis de injeção.

Os métodos descritos abaixo foram desenvolvidos na API, dessa forma ela atua como núcleo do sistema, realizando o registro e remoção de dados, treinamento do Chatbot e autenticação de usuário. Todas chamadas descritas necessitam que um *token* seja enviado no cabeçalho, exceto as chamadas *talk* e *login*, esta última é a chamada onde obtém-se o token através de um usuário e senha.

- **Nome:** Lista de *feedbacks*
Caminho: /api/rating/response
Método: GET
Cabeçalho: "Authentication: Bearer <token>"
Descrição: Retorna a lista de todos os *feedbacks* (avaliação positiva ou negativa) incluindo a pergunta do usuário e a resposta do Chatbot.
- **Nome:** Adiciona *feedback*
Caminho: /api/rating/response

Método: POST

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Adicionar um registro de *feedback* (avaliação positiva ou negativa) incluindo a pergunta do usuário e a resposta do Chatbot.

- **Nome:** Remove *feedback*

Caminho: /api/rating/response/<id>

Método: DELETE

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Remove um registro de *feedback* (avaliação positiva ou negativa) incluindo a pergunta do usuário e a resposta do Chatbot.

- **Nome:** Lista de avaliações finais

Caminho: /api/rating/final

Método: GET

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Retorna a lista de avaliações finais dos usuários.

- **Nome:** Adiciona uma avaliação final

Caminho: /api/rating/final

Método: POST

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Adiciona uma avaliação final.

- **Nome:** Remove uma avaliação final

Caminho: /api/rating/final

Método: DELETE

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Remove um registro de avaliação final.

- **Nome:** Lista de perguntas e respostas para treinamento

Caminho: /api/training

Método: GET

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Retorna a lista de perguntas e respostas utilizadas para treinamento do Chatbot.

- **Nome:** Adiciona uma pergunta e resposta para treinamento

Caminho: /api/training

Método: POST

Cabeçalho: "Authentication: Bearer <token>"

Descrição: Adiciona uma pergunta e resposta para treinamento.

- **Nome:** Remove uma pergunta e resposta de treinamento
Caminho: /api/training
Método: DELETE
Cabeçalho: "Authentication: Bearer <token>"
Descrição: Remove um registro de pergunta e resposta de treinamento.

- **Nome:** Treinamento do Chatbot
Caminho: /api/restart
Método: GET
Cabeçalho: "Authentication: Bearer <token>"
Descrição: Realiza o recarregamento da biblioteca ChatterBot que atualiza a base de conhecimento do Chatbot com os dados cadastrados na base de treinamento.

- **Nome:** Autenticação de usuário
Caminho: /api/login
Método: POST
Descrição: Retorna um *token* temporário que possibilita a chamada nos métodos autenticados da API.

- **Nome:** Conversa
Caminho: /api/talk
Método: POST
Descrição: Conversa com o Chatbot, onde é retornada uma resposta para a pergunta enviada.

4.5.2 Serviço Chatbot Frontend

Este módulo consiste na página de bate-papo onde o usuário interage com o Chatbot. A Figura 8 mostra o resultado do desenvolvimento desta interface. Nesta página estão descritas as informações de utilização do Chatbot. O usuário necessita clicar no *widget* "Posso ajudar?" do canto inferior direito para que a caixa de bate-papo seja visualizada na tela (Figura 8 e Figura 9).

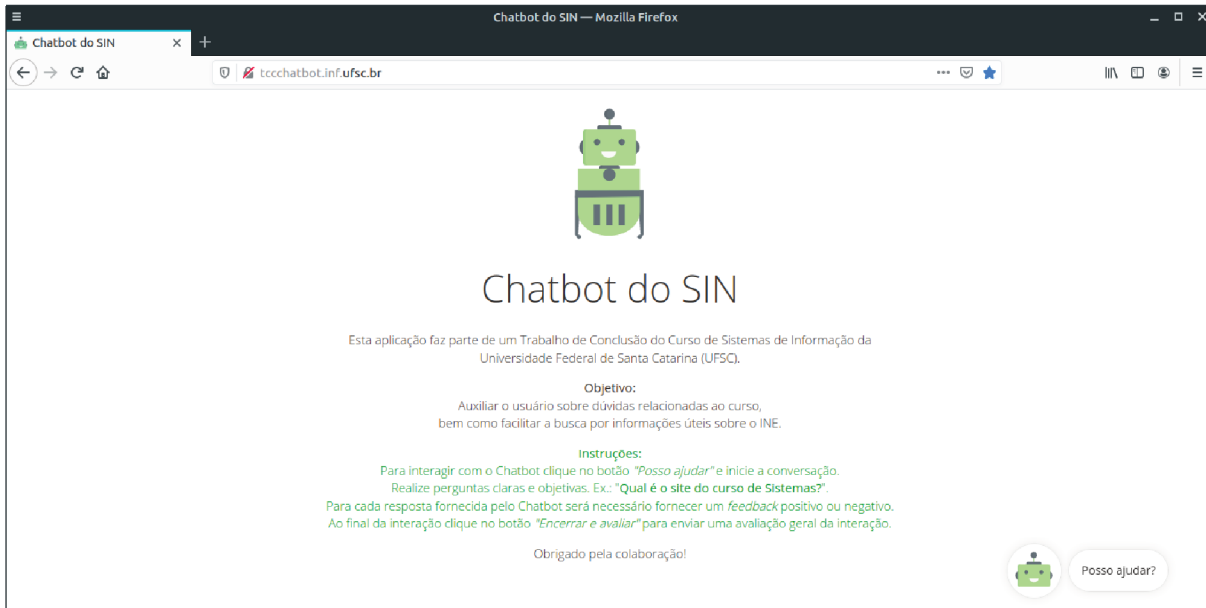


Figura 8 – Interface *web* para interação do usuário com o Chatbot.

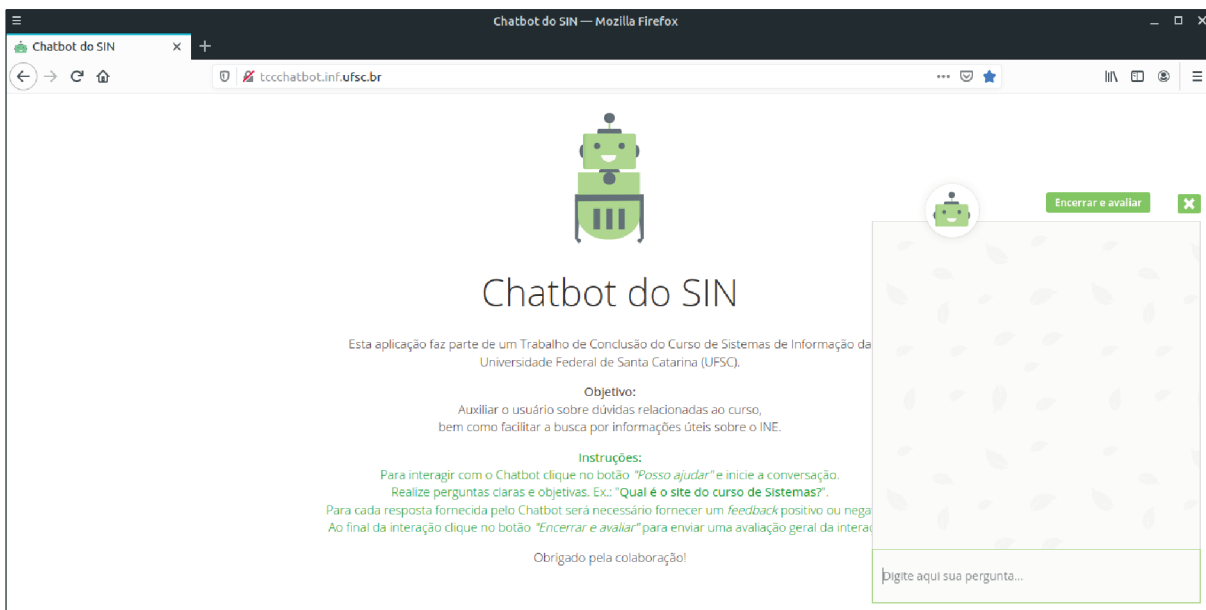


Figura 9 – Interface *web* do Chatbot com a caixa de bate-papo (*chat*) aberta.

Ao digitar uma mensagem na caixa de bate-papo, o usuário receberá uma mensagem de retorno do Chatbot. Neste momento a caixa de envio de mensagem ficará inativa até que o usuário realize o *feedback* da resposta do Chatbot, indicando se a respostas foi útil através do ícone 👍, ou se a resposta não foi útil através do ícone 👎 (Figura 10). Essa obrigatoriedade de *feedback* pode ser desativada através de uma variável de ambiente do serviço Chatbot Frontend.



Figura 10 – Interface *web* do Chatbot com formulário de envio de mensagem inativo aguardando o *feedback* do usuário.

Ao final da interação com o Chatbot, o usuário tem a opção de realizar uma avaliação final, para isso o botão "Encerrar e avaliar" deve ser pressionado (Figuras 11 e 12). Esse evento abre uma caixa de avaliação na parte superior do bate-papo, possibilitando o usuário selecionar o número de estrelas (ícone ★) que representa a utilidade geral das respostas recebidas pelo Chatbot.

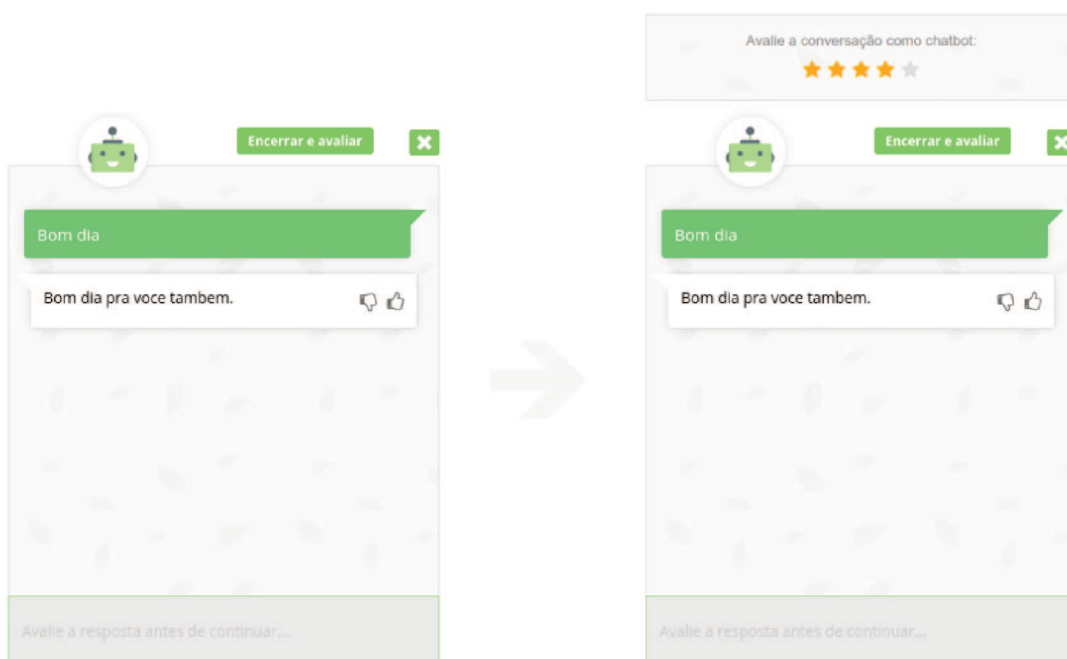


Figura 11 – Interface *web* para interação do usuário com o Chatbot.



Figura 12 – Interface *web* do Chatbot com a caixa de avaliação final aberta.

A interface do bate-papo foi desenvolvida para apresentar um comportamento semelhante independente do tamanho da tela utilizada para o acesso. Essa responsividade do bate-papo faz com que ele possa ser acessado tanto via navegador de um computador *Desktop* como de um navegador de dispositivo móvel como um *smartphone*. A Figura 13 mostra um acesso realizado com *smartphone* utilizando sistema operacional *Android*.

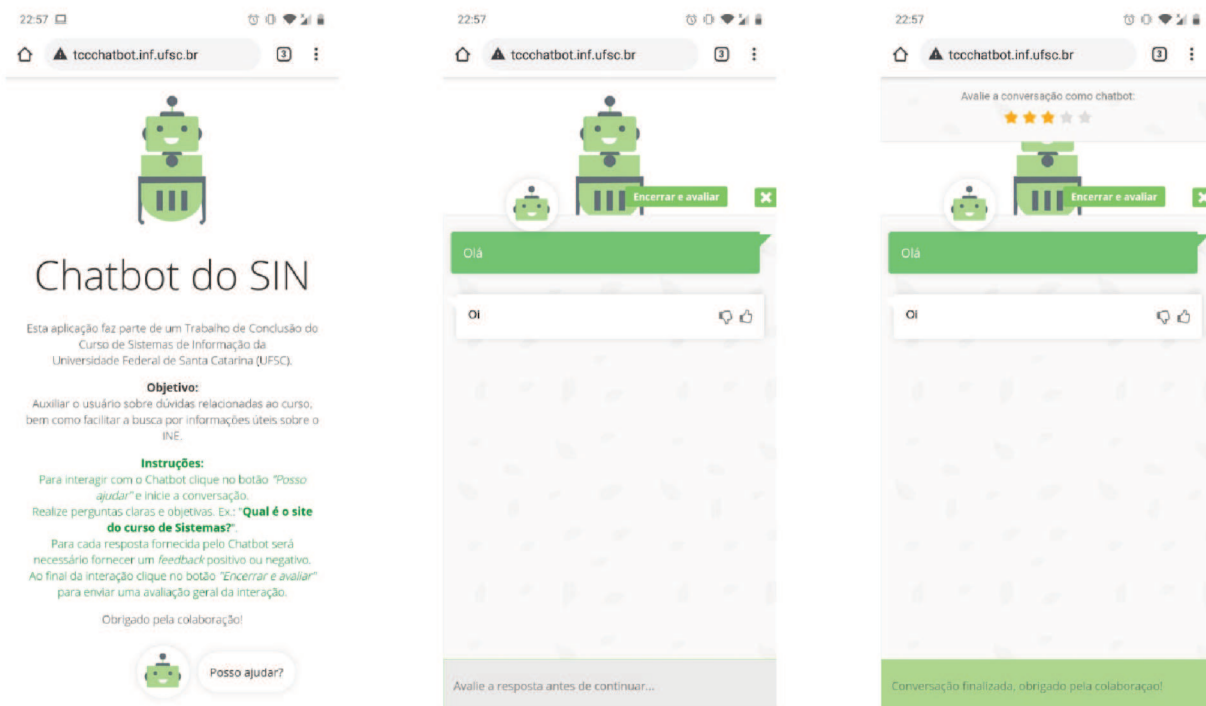


Figura 13 – Chatbot Frontend acessado via *Smartphone*.

4.5.3 Serviço Chatbot Admin

A área administrativa do sistema, denominada Chatbot Admin, foi construída utilizando-se a tecnologia *Node.js*, que é um *software* de código aberto que possibilita a execução de códigos *JavaScript* tanto no backend (servidor) como no frontend (navegador do usuário). Esse serviço foi desenvolvido utilizando-se um conjunto de bibliotecas de código livre. Dentre elas, destacam-se o *framework* para aplicações *web* *Express.js*, que possibilitou o desenvolvimento do backend, além da biblioteca *Handlebars* que atua como um processador de templates para geração do conteúdo estático de forma dinâmica.

O acesso à área administrativa necessita de autenticação através de um usuário e senha, mantendo uma sessão ativa por um determinado tempo. A página de autenticação é mostrada na Figura 14. Para manter essa sessão ativa foi utilizada a biblioteca *cookie-session*, a qual mantém os dados da sessão codificados e armazenados no cookie do usuário autenticado.

Após a autenticação com usuário e senha válidos, o usuário administrador é direcionado à página principal da área administrativa (Figura 15). Esta página disponibiliza informações sobre o teor do sistema e a descrição de cada uma das seções dessa área, que podem ser acessadas através do menu superior.

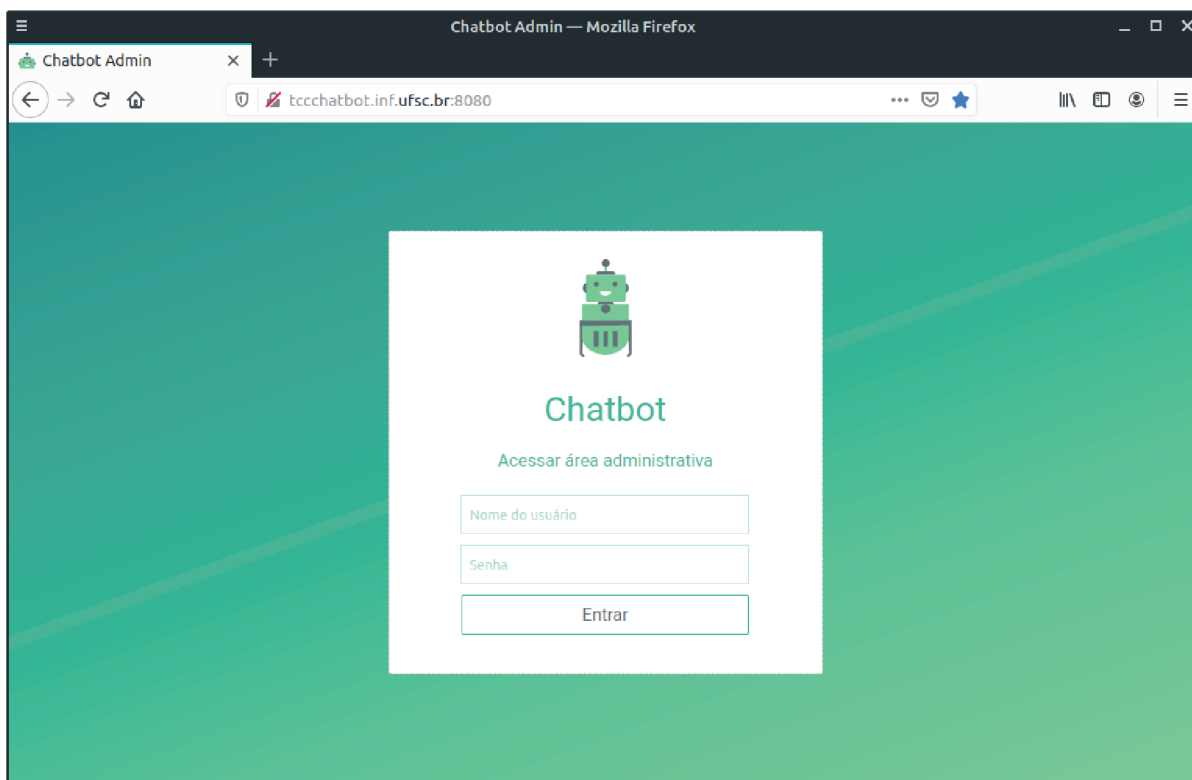


Figura 14 – Interface *web* de autenticação para acesso à área administrativa.

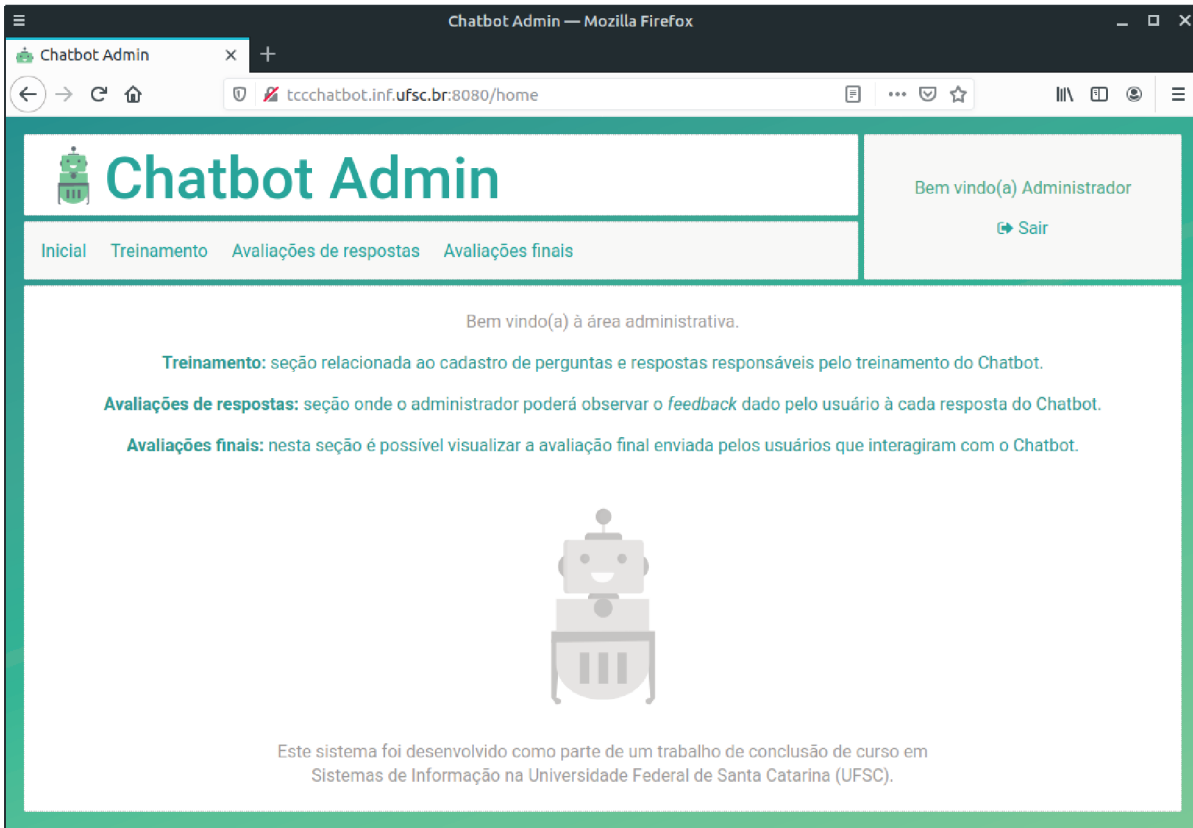


Figura 15 – Página inicial da interface *web* da área administrativa.

Na seção "Treinamento", o administrador tem acesso às perguntas e respostas cadastradas. Ao selecionar a opção "Treinar Chatbot", o sistema irá utilizar todas as perguntas e respostas cadastradas nesta seção para treinar a biblioteca ChatterBot. Uma mensagem informando o término do treinamento é mostrado ao usuário. Assim, o administrador pode cadastrar novas perguntas e respostas acionando o botão "Criar nova pergunta". Uma vez criada a pergunta e resposta, é necessário treinar novamente o Chatbot para que essas novas informações sejam adicionadas à base de conhecimento do Chatbot.

A Figura 16 mostra as funcionalidade da interface, onde é possível selecionar o número de resultados visíveis (A), ordenar por coluna (B), pesquisar algum termo (C), excluir um registro (D) e navegar pela paginação (C).

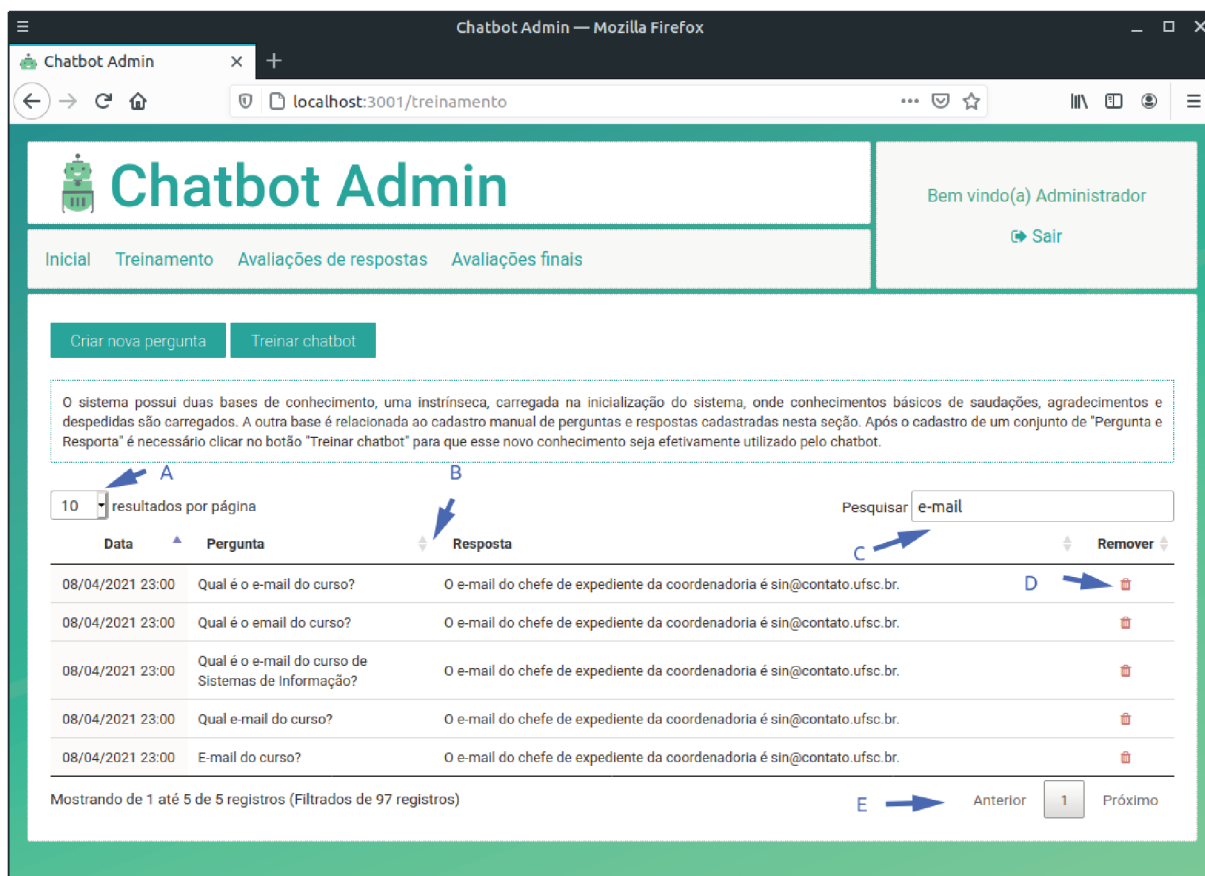


Figura 16 – Funcionalidades da interface *web*. Número de resultados visíveis (A), ordenação por coluna (B), pesquisa (C), exclusão de registro (D) e paginação (E)

O cadastro de novas perguntas e resposta é realizado por meio de um formulário (Figura 17). Neste formulário é possível adicionar mais de uma pergunta para a mesma resposta. Isto é importante para aumentar a assertividade do Chatbot. Uma vez que a biblioteca é capaz de reconhecer sentenças semelhantes, com um grau de similaridade ajustável, ao adicionar variações de perguntas que possam ser respondidas pela mesma resposta, a assertividade do Chatbot é aumentada. Por exemplo, ao cadastrar perguntas como "Qual é o telefone da secretaria do curso de Sistemas de Informação?" e "Gostaria de saber o telefone do curso de Sistemas de Informação?", a mesma resposta pode ser aplicada para as duas perguntas. Desta forma, a biblioteca irá reconhecer as frases idênticas e também frases um grau de similaridade, por exemplo, será capaz de responder a pergunta "Qual é o telefone da secretaria do curso?".

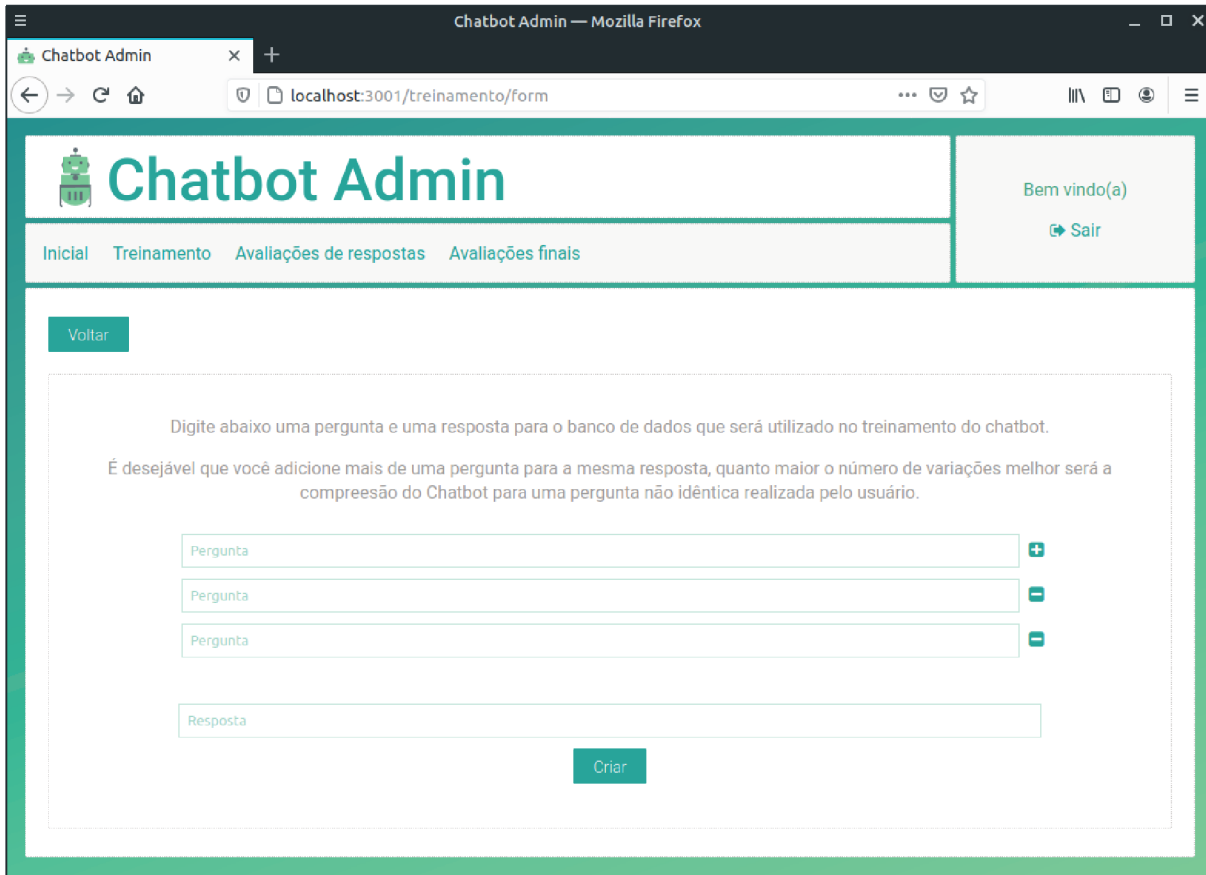


Figura 17 – Interface *web* - cadastro de novos conhecimentos através de pergunta(s) e resposta.

Com o objetivo de avaliar as interações entre o Chatbot e os usuários, a seção "Avaliações de respostas" mostra registro de *feedback* para um determinado par de pergunta e resposta (Figura 18). Dessa forma, é possível avaliar os *feedbacks* com base na pergunta e resposta, auxiliando o administrador no cadastro de novos conhecimentos para o Chatbot.

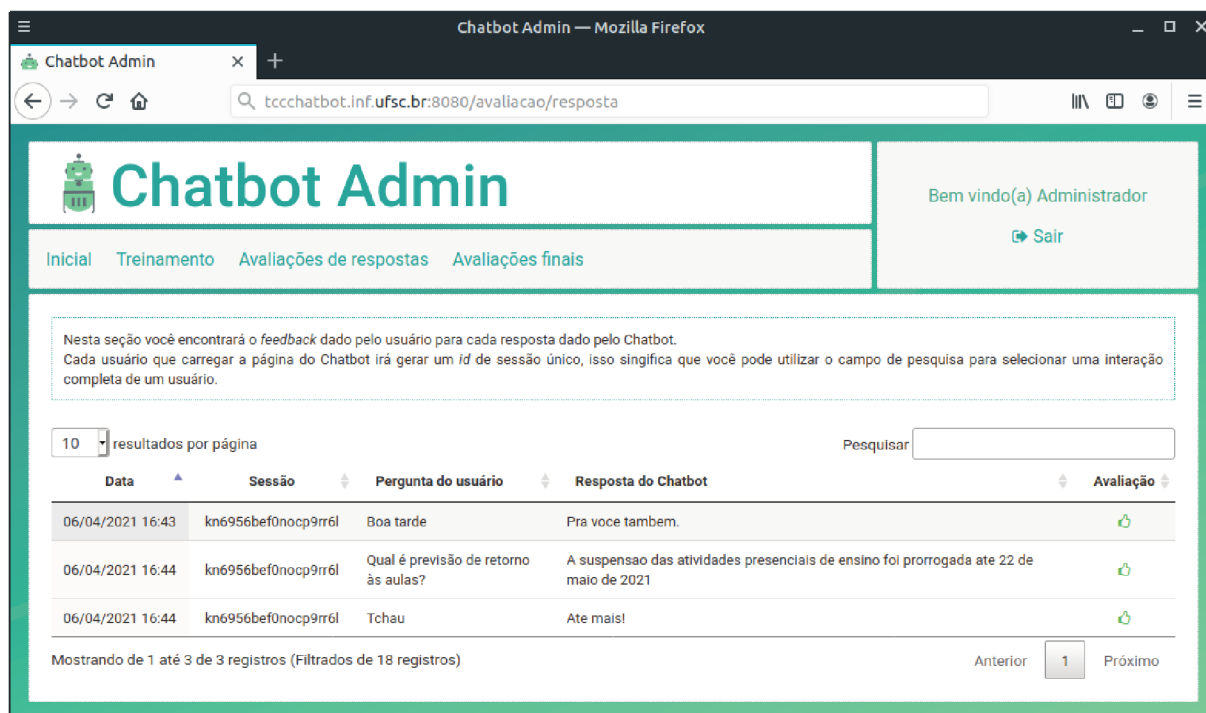


Figura 18 – Área de visualização de *feedbacks* para as respostas fornecidas pelo Chatbot.

A página de avaliações finais (Figura 20) mostra a avaliação registrada por um usuário em determinada sessão. Em cada acesso ao serviço Chatbot Frontend, um id de sessão único é gerado, caso o usuário recarregue a página, um novo id é gerado. Este id de sessão é enviado sempre que um *feedback* ou uma avaliação final sejam enviados pelo usuário. Isto possibilita o administrador buscar as informações de determinada avaliação na seção de respostas (Figura 19). Ou seja, se o administrador identificar uma avaliação de apenas uma estrela (muito insatisfeito), ele pode copiar esse id e buscar todas as perguntas e respostas na aba "Avaliação de respostas", isso o ajudará a criar perguntas mais assertivas para o conhecimento do Chatbot com base nessa interação não satisfatória.

As funcionalidades das seções "Avaliações de respostas" e "Avaliações finais" são semelhantes às descritas para a seção de "Treinamento", com a capacidade de paginação, busca, ordenação e seleção de resultados por página.

Chatbot Admin — Mozilla Firefox

Chatbot Admin

tcccchatbot.inf.ufsc.br:8080/avaliacao/resposta

Chatbot Admin

Bem vindo(a) Administrador

Sair

Inicial Treinamento Avaliações de respostas Avaliações finais

Nesta seção você encontrará o feedback dado pelo usuário para cada resposta dado pelo Chatbot. Cada usuário que carregar a página do Chatbot irá gerar um *id* de sessão único, isso significa que você pode utilizar o campo de pesquisa para selecionar uma interação completa de um usuário.

10 resultados por página

Pesquisar

Data	Sessão	Pergunta do usuário	Resposta do Chatbot	Avaliação
13/04/2021 23:45	knqoadjqhgdfmebfr9	Boa noite	Boa noite pra voce tambem.	👍
13/04/2021 23:46	knqoadjqhgdfmebfr9	Qual o objetivo do curso de Sistemas?	O curso de Sistemas tem como objetivo formar profissionais que contribuam para a solucao de problemas de tratamento de informacao.	👍
13/04/2021 23:46	knqoadjqhgdfmebfr9	Obrigado	Disponha!	👍

Mostrando de 1 até 3 de 3 registros (Filtrados de 24 registros)

Anterior 1 Próximo

Figura 19 – Funcionalidade de pesquisa de mensagens trocadas entre o usuário e o Chatbot para o id de sessão *knqoadjqhgdfmebfr9*.

Chatbot Admin — Mozilla Firefox

Chatbot Admin

tcccchatbot.inf.ufsc.br:8080/avaliacao/final

Chatbot Admin

Bem vindo(a) Administrador

Sair

Inicial Treinamento Avaliações de respostas Avaliações finais

Nesta seção você encontrará o feedback final do usuário, ou seja, a nota de 1 a 5 avaliando o completo diálogo. Cada usuário que carregar a página Com o *id* da sessão é possível navegar até o menu "Avaliações de respostas" e visualizar todo o diálogo da interação.

10 resultados por página

Pesquisar

Data	Usuário (sessão)	Avaliação
06/04/2021 16:44	kn6956bef0nocp9rr6l	★★★★★
06/04/2021 17:04	kn69uf2pyegI5sev13h	★★★★☆
11/04/2021 15:03	kndargnyvz38l24tyy	★★★★☆
11/04/2021 19:29	kndk9jadn6hfc84t1rg	★★★☆☆

Mostrando de 1 até 4 de 4 registros

Anterior 1 Próximo

Figura 20 – Área de visualização das avaliações finais de interação com o Chatbot.

5 Resultados e Discussões

Nas seções anteriores foram elencados os requisitos funcionais e não funcionais do sistema, as regras de negócio, a arquitetura proposta, tecnologias para cada serviço, funcionalidades em diagramas de atividades e o modelo do banco de dados, bem como informações de implementação do sistema. Com base nesses dados, o sistema foi implementado e disponibilizado através de um endereço eletrônico com o objetivo de ser avaliado por alunos do curso de Sistemas de Informação da UFSC. A disseminação do endereço do Chatbot foi realizada através do envio de um e-mail pela coordenação do curso, no qual, usuários voluntariamente interagiram com o Chatbot. Os resultados discutidos nesta seção representam interações que ocorreram em um período de seis dias de exposição do sistema.

A Tabela 3 mostra o número total de mensagens enviadas ao Chatbot, totalizando 394. Deste total, 271 foram consideradas perguntas válidas e 123 foram consideradas inválidas. O conjunto considerado inválido foi representado por mensagens que não representam uma pergunta propriamente dita, tentativas de invasão ao sistema, tentativas de inativação do sistema, mensagens ofensivas e mensagens com erros ortográficos e gramaticais.

Tabela 3 – Total de perguntas e *feedbacks* negativos e positivos realizados no Chatbot.

Perguntas válidas		271
Avaliações positivas	140	
Avaliações negativas	131	
Perguntas inválidas		123
Total de perguntas		394

Em relação aos *feedbacks* realizados pelos usuários a cada respostas do Chatbot (Tabela 3), obteve-se um número de 140 (51,7%) para avaliações positivas e 131 (48,3%) para avaliações negativas. Ao avaliar individualmente os *feedbacks* negativos, observou-se que muitas perguntas direcionadas ao Chatbot não possuem respostas por não se tratarem do contexto de respostas do Chatbot. Uma vez que o objetivo da aplicação é auxiliar sobre questões pertinentes ao curso de Sistemas de Informação da UFSC. Estas perguntas não foram consideradas inválidas, uma vez que podem ser incluídas na base de conhecimento do Chatbot com o objetivo de torná-lo mais amigável e atrativo ao usuário. Por exemplo, foram observadas perguntas como: "Por que o céu é azul?", "O certo é biscoito ou bolacha?", "Qual o valor da coxinha?", "Qual o cardápio do restaurante universitário", entre outros.

O total de sessões abertas no navegador que interagiram com o Chatbot foi 83. Dentre este total, apenas 33 realizaram a avaliação final da interação. Isto significa que nem todos os usuários que interagiram com o Chatbot realizaram a avaliação final. A Figura 21 mostra o resultado do julgamento dos usuário que realizaram a avaliação selecionando uma das opções de valor, de 1 estrela (muito insatisfeito) até 5 estrelas (muito satisfeito).

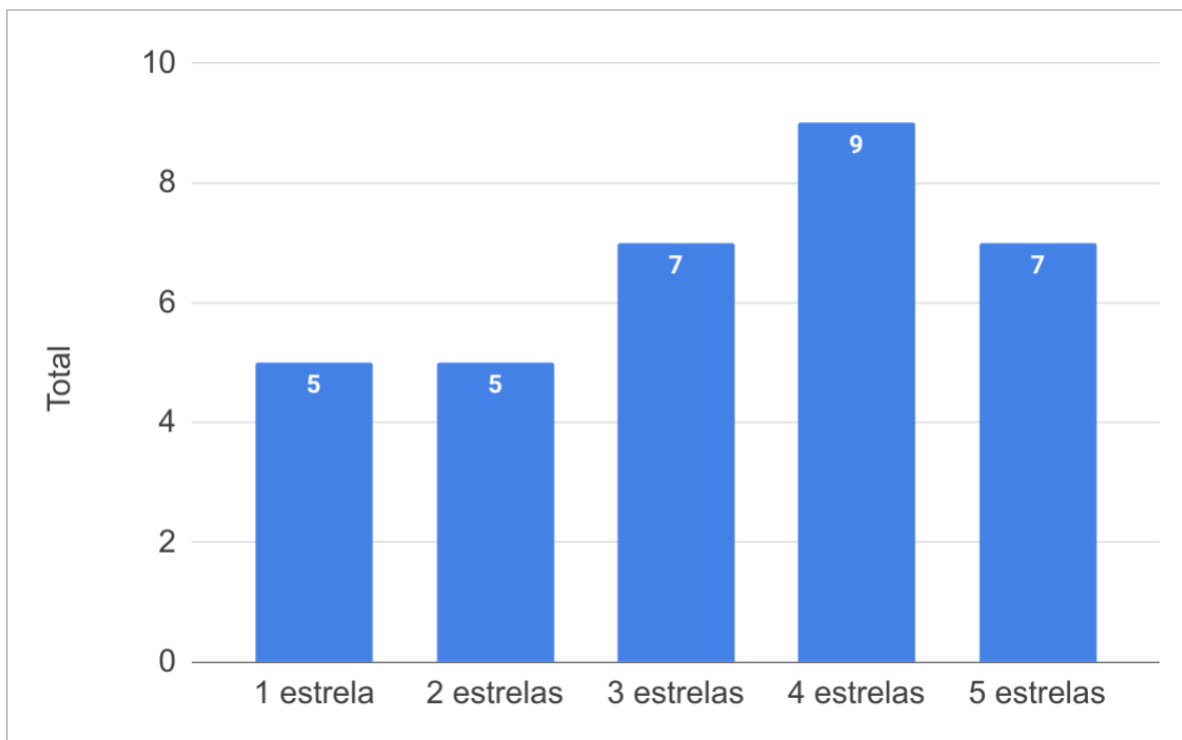


Figura 21 – Avaliações finais registradas pelos usuários.

O número total de perguntas e respostas utilizado no treinamento do Chatbot para essa primeira exposição a usuários reais foi de 150. Este é um valor ainda pequeno para um Chatbot poder fornecer uma variedade de respostas e abranger todo o domínio de conhecimento pertinente às questões do curso de Sistemas de Informação. O objetivo dessa primeira interação com usuários foi realizar um levantamento sobre os principais assuntos de interesse dos possíveis usuários do Chatbot.

A Tabela 4 mostra a lista dos assuntos mais solicitados ao Chatbot. O grupo "Genérico" é representado por saudações, agradecimentos, despedidas e questionamentos fora do contexto de Sistemas de Informação. O segundo assunto mais evidenciado foi questionamentos sobre o Chatbot, por exemplo, nome, idade, humor, funcionalidades, etc. Em seguida é possível observar assuntos pertinentes ao curso, como questionamentos sobre a matrícula, contatos da coordenadoria e do coordenador, bem como a duração do curso, informações sobre o site, grade de horários, estágio, localização física e trancamento de matrícula.

Tabela 4 – Principais assuntos questionados ao Chatbot.

Número de ocorrências	Assunto	Descrição
89	Genérico	Saudações, agradecimento, despedidas, mensagens fora do contexto.
19	Sobre o Chatbot	Perguntas sobre o Chatbot, como por exemplo: nome, idade, humor.
11	Coordenador do curso	Perguntas sobre quem é o coordenador do curso.
11	Matrícula	Questões sobre realização da matrícula, localização, documentos.
10	Contato da coordenadoria	Contatos da coordenadoria, como telefone e e-mail.
10	Duração do curso	Tempo total de duração do curso em anos e semestres.
7	Site do curso	Solicitações de endereço eletrônico do curso.
7	Grade de horários	Questionamentos gerais sobre grade de horários e a sua localização.
6	Estágio	Questionamentos gerais sobre estágio, duração, alteração, orientação.
6	Localização do SIN/INE	Endereço eletrônico e físico do INE/SIN
5	Trancamento de matrícula	Dúvidas sobre como realizar o trancamento de matrícula e o período permitido.
5	Professor de uma disciplina	Questionamento sobre o nome do professor de determinada disciplina.

Com base no número de ocorrências de perguntas em determinados assuntos é possível, através do painel administrativo, cadastrar novos conhecimentos utilizando esta informação como um guia para aumentar a assertividade do Chatbot melhorando assim a utilidade da ferramenta.

6 Conclusão

O presente trabalho atingiu o objetivo geral criando um sistema do tipo Chatbot com base em uma biblioteca de código aberto. Sobre essa biblioteca foram desenvolvidas camadas do sistema que resultaram em uma aplicação capaz de interagir com um usuário no formato de bate-papo, utilizando a linguagem natural através de uma interface *web*. Além de uma área administrativa para visualizar as interações e, com base nessas informações, criar perguntas e respostas para treinamento do sistema aumentando sua base de conhecimento.

O levantamento dos requisitos funcionais e não funcionais, bem como as regras de negócio delinearão a construção do sistema, a qual fundamentou-se em linguagens e tecnologias de código aberto. Com base nesses dados, foi desenvolvido o sistema proposto nos objetivos do trabalho.

A interface *web* de bate-papo, serviço "Chatbot Frontend", mostrou-se suficiente para permitir a interação dos usuários através de linguagem natural em formato de mensagem de texto, uma vez que não foi encontrada nenhuma dificuldade por parte dos usuários no envio de mensagens para o Chatbot. Assim como a área administrativa, "Serviço Admin", mostrou-se eficiente para administrar o Chatbot, através das funcionalidades básicas de Treinamento e Avaliações. No entanto, sugere-se com trabalho futuro a validação desta área juntamente com a coordenadoria e secretaria do curso de Sistemas de Informação.

O levantamento de dados, descrito na seção de resultados, sobre os principais assuntos abordados pelos usuários, fornece uma base para auxiliar o administrador no cadastro de novas perguntas e respostas, de forma direcionada aos assuntos mais solicitados. Juntamente com o gerenciamento das informações observadas nos menus de "Avaliações finais" e "Avaliações de respostas" do painel administrativo, fornecem uma base para direcionar o usuário administrador sobre quais as perguntas são pertinentes aos alunos do curso, evidenciando também quais não são contempladas pelo conhecimento atual do Chatbot, necessitando ser incorporadas na base de dados. Desta forma, o acréscimo da base de conhecimento proporcionará uma melhora na qualidade das respostas tornando a ferramenta cada vez mais útil na busca por informações sobre o curso de Sistemas de Informação, tanto para alunos do curso como para pessoas externas à ele, que buscam alguma informação de forma ágil.

O resultado das avaliações finais mostra que as interações obtiveram resultados variados de satisfação, de muito insatisfeito até muito satisfeito. Isto mostra a dificuldade em se trabalhar com um sistema que interage com usuários sem regras claras de comunicação. A linguagem utilizada pelos usuários compreendeu desde palavras abreviadas, gírias até a descontextualização do objetivo da interação. Levando-se em conta que foi realizado

apenas um ciclo de interação com usuários reais, a base de conhecimento do Chatbot pode evoluir para uma assertividade maior. Uma vez que, as mensagens coletadas no primeiro ciclo servem de matéria-prima para o cadastro e treinamento de novas perguntas e respostas. Portanto, o gerenciamento da base de conhecimento deve ser incremental com o intuito de aumentar em relação ao tamanho e assertividade.

Na seção seguinte são sugeridos pontos importantes que podem ser incorporados no sistema de Chatbot a fim de melhorar a qualidade e utilidade da ferramenta.

6.1 Trabalhos futuros

Durante a etapa de desenvolvimento e avaliação dos resultados, foram observados alguns pontos que podem melhorar o sistema de Chatbot e trazer mais benefícios ao administrador ou, uma maior satisfação para o usuário final.

Com o objetivo de tornar o Chatbot mais amigável é oportuno criar uma base de conhecimento para responder perguntas que usuários costumam fazer fora do contexto do curso de Sistemas de Informação. Essas perguntas e respostas podem ser adicionadas às bases de conhecimento que são carregadas na inicialização do sistema, como a base de conhecimento para agradecimentos, despedidas e saudações.

Através das interações dos usuários com o Chatbot, observou-se tentativas de invasão ao sistema. Essas tentativas não afetaram o banco de dados devido à utilização de um ORM que impede a injeção de SQL malicioso. No entanto, as requisições para os métodos que não necessitam de *token* são passíveis de falhas de segurança. Neste sentido, é importante melhorar as validações de dados destas chamadas, a fim de verificar o tipo e quantidade de dados enviado ao servidor. Outro ponto de segurança importante seria verificar a procedência das requisições à API, limitando apenas para origens conhecidas, no caso o serviço Chatbot Frontend e Admin.

A área administrativa realiza requisições ao Chatbot API solicitando as informações do banco de dados, essas informações não são respondidas de forma paginada. Isto não é um problema em um curso inicial da aplicação, porém, com o passar do tempo isto pode tornar-se custoso computacionalmente tanto ao servidor como na renderização no navegador do usuário administrativo. Portanto, uma oportunidade de melhoria do sistema seria a paginação no lado do servidor.

Também na área administrativa, na seção de treinamento, não existe a funcionalidade de edição de uma pergunta/resposta, sendo necessário remover o registro e inserir um novo em caso de uma inserção incorreta. Ainda, nas seções de "Avaliações Finais" e "Avaliações de Respostas" não existe a funcionalidade de remoção de registros, a qual poderia ser útil após a sua utilização para supervisão. Portanto, estas funcionalidades

poderiam ser adicionadas à área administrativa, tornando-a mais eficiente e dinâmica.

Apesar das melhorias elencadas para a área administrativa, a validação desse módulo do sistema poderia ser realizada em conjunto com a coordenadoria e secretaria do curso, desta forma, melhorias mais eficientes poderiam ser identificadas.

Uma forma de melhorar o treinamento do sistema poderia ser realizada por meio da incorporação de um pré-processamento dos conjuntos de perguntas e respostas cadastrados no painel administrativo. Esse pré-processamento poderia ser realizado com a utilização de ferramentas de PLN que utilizem técnicas de *stemming*, lematização, entre outras. Desta forma, ao cadastrar uma pergunta, poderiam ser geradas variações desta pergunta de forma automática, aumentando a capacidade de interpretação do Chatbot.

Finalmente, como uma forma de melhorar a compreensão das mensagens dos usuários, sugere-se que o Chatbot tenha a funcionalidade de interpretar, até certo ponto, palavras digitadas incorretamente. Por exemplo, o usuário pode digitar apenas uma letra de palavra incorretamente e não evidenciar o erro, o sistema verificaria a inexistência da palavra e buscaria em um dicionário uma nova palavra com um índice mínimo de similaridade.

Referências

- AKBIK, A.; BLYTHE, D.; VOLLGRAF, R. Contextual string embeddings for sequence labeling. In: *COLING 2018, 27th International Conference on Computational Linguistics*. [S.l.: s.n.], 2018. p. 1638–1649. Citado na página 21.
- BIRD EDWARD LOPER, E. K. S. *NLTK*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://nltk.org>>. Citado na página 21.
- COX, G. *ChatterBot: Machine learning in Python*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://chatterbot.readthedocs.io>>. Citado 2 vezes nas páginas 22 e 31.
- DAHL, R. *Node.js*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://flask.palletsprojects.com>>. Citado na página 32.
- FINCH, S. E. et al. Emora: An inquisitive social chatbot who cares for you. *arXiv preprint arXiv:2009.04617*, 2020. Citado na página 21.
- FRENCH, R. M. The turing test: the first 50 years. *Trends in cognitive sciences*, Elsevier, v. 4, n. 3, p. 115–122, 2000. Citado na página 23.
- GOLDBERG, Y. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017. Citado na página 20.
- HAPKE, H. M.; LANE, H.; HOWARD, C. *Natural language processing in action*. [S.l.]: Manning, 2019. Citado na página 19.
- HONNIBAL, M.; MONTANI, I. *spaCy*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://spacy.io>>. Citado na página 21.
- HOSSAIN, M. M. et al. Auto-correction of english to bengali transliteration system using levenshtein distance. In: IEEE. *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. [S.l.], 2019. p. 1–5. Citado na página 20.
- HYKES, S. *Docker*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://www.docker.com>>. Citado na página 33.
- INTELLIGENCE, B. I. *MS Windows NT 80% of businesses want chatbots by 2020*. 2016. Disponível em: <<https://www.businessinsider.com/80-of-businesses-want-chatbots-by-2020-2016-12>>. Citado na página 23.
- KACZOROWSKA-SPYCHALSKA, D. Chatbots in marketing. *Management*, v. 23, n. 1, 2019. Citado na página 23.
- KASHEFI, O.; SHARIFI, M.; MINAIE, B. A novel string distance metric for ranking persian respelling suggestions. *Natural Language Engineering*, Cambridge University Press, v. 19, n. 2, p. 259, 2013. Citado na página 20.
- KOWSHER, M. et al. Doly: Bengali chatbot for bengali education. In: IEEE. *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. [S.l.], 2019. p. 1–6. Citado na página 21.

- LORIA, S. *TextBlob*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://textblob.readthedocs.io>>. Citado na página 21.
- MASLOWSKI, I.; LAGARDE, D.; CLAVEL, C. In-the-wild chatbot corpus: from opinion analysis to interaction problem detection. In: *International conference on natural language and speech processing*. [S.l.: s.n.], 2017. Citado na página 21.
- MAULDIN, M. L. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In: *AAAI*. [S.l.: s.n.], 1994. v. 94, p. 16–21. Citado na página 23.
- MOPPEL, J. *Socratic chatbot*. [S.l.]: University Of Tartu, Institute of Computer Science, Bachelor’s Thesis, 2018. Citado na página 21.
- NEUHAUSER, L.; KREPS, G. L. Participatory design and artificial intelligence: Strategies to improve health communication for diverse audiences. In: *2011 AAAI Spring Symposium Series*. [S.l.: s.n.], 2011. Citado na página 13.
- RAHANE, W. et al. Artificial intelligence based solarbot. In: IEEE. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. [S.l.], 2018. p. 601–605. Citado na página 13.
- RAHMAN, A.; MAMUN, A. A.; ISLAM, A. Programming challenges of chatbot: Current and future prospective. In: IEEE. *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. [S.l.], 2017. p. 75–78. Citado na página 14.
- RONACHER, A. *Flask*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://flask.palletsprojects.com>>. Citado na página 32.
- ROY, T. F. The representational state transfer (rest). *Department of Information and Computer Science, UCI*, 2000. Citado na página 32.
- RUF, B. et al. Pharmabroad: a companion chatbot for identifying pharmaceutical products when traveling abroad. In: *Information and Communication Technologies in Tourism 2020*. [S.l.]: Springer, 2020. p. 218–228. Citado na página 21.
- RUSSEL, S.; NORVIG, P. Inteligência artificial. 3a edição. *Editora Campus*, 2013. Citado na página 14.
- SANDERS, N. C.; CHIN, S. B. Phonological distance measures. *Journal of quantitative linguistics*, Taylor & Francis, v. 16, n. 1, p. 96–114, 2009. Citado na página 20.
- SINGH, P. *Machine Learning with PySpark: With Natural Language Processing and Recommender Systems*. [S.l.]: Apress, 2018. Citado na página 21.
- TURING, A.; HAUGELAND, J. *Computing machinery and intelligence*. [S.l.]: MIT Press Cambridge, MA, 1950. Citado na página 23.
- WEIZENBAUM, J. et al. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, New York, NY, USA, v. 9, n. 1, p. 36–45, 1966. Citado na página 13.
- WIDENIUS, U. M. *MariaDB Foundation*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://mariadb.org>>. Citado na página 33.

XIE, Q. et al. Chatbot application on cryptocurrency. In: IEEE. *2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFER)*. [S.l.], 2019. p. 1–8. Citado na página [25](#).

Apêndices

APÊNDICE A – Arquivo YAML com as instruções para o orquestrador de contêineres Docker compose.

```
version: "3"
services:

  ## CHATBOT-FRONTEND =====
  chatbot_frontend:
    container_name: chatbot_frontend
    image: node:15.10.0-alpine3.10
    user: root
    ports:
      - "3000:8080"
    volumes:
      - ./chatbot_frontend:/home/node/app
    command: sh -c "cd /home/node/app && yarn install && yarn run start-dev"
    environment:
      CHATBOT_API_URL: http://localhost:3002
      FORCE_RESPONSE_RATING: 'true'
      PORT: 8080

  ## CHATBOT-ADMIN =====
  chatbot_admin:
    container_name: chatbot_admin
    image: node:15.10.0-alpine3.10
    user: "node"
    working_dir: /home/node/app
    ports:
      - "3001:8080"
    volumes:
      - ./chatbot_admin:/home/node/app
    command: npm run start-dev
    depends_on:
      - "chatbot_api"
```

```
environment:
  PORT: 8080
  NODE_ENV: development
  CHATBOT_API_URL: http://localhost:8080
  SESSION_COOKIE_KEY1: '<hash-aleatoria-forte-1>'
  SESSION_COOKIE_KEY2: '<hash-aleatoria-forte-2>'
  SESSION_COOKIE_KEY3: '<hash-aleatoria-forte-3>'
```

```
## CHATBOT-API ==++=====
```

```
chatbot_api:
  container_name: chatbot_api
  image: pereirasthiago/chatterbot:1.1.0a7
  ports:
    - "3002:8080"
  volumes:
    - ./chatbot_api:/app
  command: python app.py
  depends_on:
    - "chatbot_db"
  environment:
    MINIMUM_CONFIDENCE: 0.6
    MYSQL_HOST: chatbot_db
    MYSQL_USER: chatbot_app
    MYSQL_PASSWORD: <senha-forte>
    MYSQL_DATABASE: chatbot
    MYSQL_PORT: 3306
    FLASK_ENV: development
    ADMIN_NAME: 'Administrador'
    ADMIN_USERNAME: 'admin'
    ADMIN_PASSWORD: '<senha-do-administrador>'
    SECRET_KEY: '<hash-aleatoria-forte>'
    JWT_SECRET_KEY: '<hash-aleatoria-forte>'
    JWT_ACCESS_TOKEN_EXPIRES_MINUTES: 1440
    PORT: 8080
```

```
## CHATBOT-DB =====
```

```
chatbot_db:
  container_name: chatbot_db
```

```
image: jbergstroem/mariadb-alpine:10.4.17
restart: always
ports:
  - "3306:3306"
volumes:
  - ${PWD}/.mariadb:/var/lib/mysql
environment:
  MYSQL_ROOT_PASSWORD: <senha-do-root>
  MYSQL_USER: app_chatbot
  MYSQL_PASSWORD: <senha-app-chatbot>
  MYSQL_DATABASE: chatbot
```


APÊNDICE B – Código Dockerfile

desenvolvido para a imagem

chatterbot:1.1.0a7.

```
FROM python:3.7.9-alpine3.13

WORKDIR /app

ENV WORKDIR=/app

RUN apk add --no-cache --virtual=build_deps g++ && \
    pip install --no-cache-dir spacy==2.1.8 && \
    pip install --no-cache-dir ChatterBot==1.1.0a7 && \
    pip install --no-cache-dir PyYAML==5.3.1 && \
    pip install --no-cache-dir chatterbot-corpus==1.2.0 && \
    python -m spacy download en && \
    python -m spacy download pt && \
    pip install --no-cache-dir Flask==1.1.2 && \
    pip install --no-cache-dir PyMySQL==1.0.2 && \
    pip install --no-cache-dir Flask-Cors==3.0.10 && \
    apk add --no-cache libstdc++ && \
    pip install --no-cache-dir flask-marshmallow==0.14.0 && \
    pip install --no-cache-dir Flask-SQLAlchemy==2.4.4 && \
    pip install --no-cache-dir marshmallow-sqlalchemy==0.24.2 && \
    pip install --no-cache-dir Flask-JWT-Extended==4.0.2 && \
    apk add --no-cache python3-dev libffi-dev && \
    pip install --no-cache-dir Flask-Bcrypt==0.7.1 && \
    apk del g++ python3-dev libffi-dev && \
    apk del build_deps

CMD ["sh"]
```


APÊNDICE C – Artigo

Desenvolvimento de um sistema do tipo Chatbot para o curso de Sistemas de Informação

Thiago Pereira da Silva¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis, Brasil

thiago.ps@posgrad.ufsc.br

Abstract. *Human-machine interaction is increasingly present today, thereby, supply and exchange of information are being more and more automated. However, human and machine communication in natural language still face several challenges. In addition, there are several computer libraries of human-machine conversation, however, the majority are libraries to be incorporated as a part of the system. In other words, several layers are necessary so that the system can be used by users and managed by a system administrator. The aim of the present work was to develop a Chatbot system with a complete solution, from natural language processing adapted to the Portuguese language, using an open-source library, to the chat interface and the administrative system for managing. The context of the Chatbot conversation is within the scope of the undergraduate course in Information Systems at the Federal University of Santa Catarina.*

Resumo. *A interação entre humano e máquina está cada vez mais frequente nos dias de hoje, com isso o fornecimento e troca de informação está sendo cada vez mais automatizado. No entanto, a comunicação de um usuário com uma máquina através da linguagem natural ainda encontra diversos desafios. Além disso, existem diversas bibliotecas computacionais de conversação humano-máquina, porém, em sua grande maioria são bibliotecas para serem incorporadas como uma parte do sistema, ou seja, são necessárias diversas camadas de entorno até que o sistema possa ser utilizado por usuários na web e gerenciado por um administrador de forma amigável. Dessa forma, o objetivo do presente trabalho foi desenvolver um sistema do tipo Chatbot que apresente a solução completa, desde o processamento de linguagem natural adaptado à língua portuguesa, utilizando uma biblioteca de código aberto, até a interface de bate-papo e o sistema administrativo para gerenciamento do Chatbot. O contexto da conversação do Chatbot é no âmbito do curso de graduação em Sistemas de Informação da Universidade Federal de Santa Catarina.*

1. Introdução

Em um mundo onde os meios de comunicação digital têm crescido rapidamente, ferramentas de automação têm se tornado um diferencial no mercado ao tornar os canais de comunicação mais eficientes. A interação entre o ser humano e o computador pode ocorrer de diversas formas, entre elas, a comunicação por linguagem natural (LN). A primeira aplicação computacional desenvolvida com o objetivo de simular uma conversação foi o software ELIZA, criado por Joseph Weizenbaum entre 1964 e 1966 (WEIZENBAUM et

al., 1966). Desde então, os softwares de conversação tem avançado muito, diariamente é possível entrar em contato com assistentes do tipo Chatbot em computadores ou celulares, como é o caso da Siri da Apple, Google Assistente, Alexa da Amazon, Cortana da Microsoft, Zenbot, Robô Ed, Cedrico, entre outros.

O termo Chatbot (ou Chatterbot) vem do inglês, onde *chatter* significa “conversa fiada” e *bot* é uma abreviatura de *robot*, robô em português, consistindo em uma aplicação que proporciona um diálogo entre uma máquina e um ser humano. Esse tipo de aplicação têm sido largamente utilizada em canais de comunicação. Segundo Rahane et al. (2018), chatbots podem ser úteis em fornecer serviços a clientes, apresentar recomendações de produtos, atrair consumidores para diferentes campanhas de marketing, além de muitas outras áreas de negócios.

Para que um diálogo entre homem e máquina seja possível é necessário que a aplicação interprete os dados de entrada e realize um processamento a fim de identificar a melhor resposta para o usuário. Essa tecnologia pode ser utilizada em qualquer tipo de âmbito institucional em que haja necessidade de fornecimento e/ou recebimento de informação, a qual possa ser automatizada com objetivos específicos. O principal desafio da máquina é compreender a intenção da mensagem do ser humano. Segundo Russel e Norvig (2013), a compreensão da linguagem exige a compreensão de dois aspectos, o assunto e o contexto da mensagem e não apenas a compreensão da estrutura das frases. Atualmente, existem diversas empresas que fornecem plataformas para construção de aplicações de diálogo com usuários humanos, como é o caso da IBM Watson¹, Microsoft bot², Dialogflow³, entre outras. No entanto, o principal idioma utilizado nessas ferramentas é o inglês e, na maioria dos casos, a tradução automatizada para a língua portuguesa encontra dificuldades, pois o contexto da conversa pode ser diferente da tradução literal das palavras ou pequenas frases. Segundo Rahman, Mamun e Islam (2017), os Chatbots ainda carecem de uma vasta lógica de recursos linguísticos. Consultas complexas ainda fornecem problemas de formas singulares e plurais, sinônimos e análise sentimental.

Aplicações de Chatbot também são utilizadas em portais *web* com grande quantidade de informação. A navegação por menus e diferentes níveis de páginas pode dificultar a busca por conteúdos do portal. Desta forma, uma pequena caixa de bate-papo com um Chatbot na página principal de um portal pode ter um ganho de tempo muito grande para o usuário, uma vez que o Chatbot pode responder a dúvida com uma mensagem completa ou uma URL apontando para a localização correta da informação. O portal do curso de Sistemas de Informação da UFSC (<https://sin.ufsc.br>) possui uma grande quantidade de informações, apesar da boa organização, um aplicativo de Chatbot na página inicial poderia fornecer ainda mais agilidade aos estudantes e demais usuários. Além de poder fornecer informações de forma ágil de diferentes níveis do portal, também pode conter informações que estão apenas disponíveis normalmente em arquivos do tipo PDF, documentos e até informações não disponíveis no portal.

Dessa forma, o presente trabalho tem o objetivo de desenvolver um sistema *web* de Chatbot com base em um mecanismo de conversação de código aberto. Este sistema engloba uma aplicação para processamento de linguagem natural adaptado à língua por-

¹<https://www.ibm.com/watson>

²<https://azure.microsoft.com/en-us/services/bot-service>

³<https://cloud.google.com/dialogflow>

tuguesa, uma interface de conversação e uma área administrativa. Assim, permitindo um fácil gerenciamento da base de conhecimento do Chatbot, com a possibilidade incrementar o seu banco de dados com novas informações. Além disso, o administrador será capaz de supervisionar diálogos não compreendidos pelo Chatbot, adicionando respostas às perguntas não reconhecidas visualizadas no histórico de conversas, com o objetivo de supervisionar e ensinar novos conhecimentos ao sistema. O tema empregado ao conhecimento do Chatbot é relacionado ao curso de graduação em Sistemas de Informação da Universidade Federal de Santa Catarina, nesse contexto, os usuários poderão dialogar com o Chatbot a fim de obter informações de forma rápida e interativa sobre dúvidas e questionamentos usuais, sem a necessidade de uma pesquisa mais aprofundada no portal ou contato com coordenador e chefe de expediente.

2. Metodologia

Para atingir os objetivos propostos, o presente trabalho utilizou os seguintes procedimentos metodológicos em ordem cronológica. Inicialmente, foi realizada uma revisão bibliográfica e contextualização sobre Chatbots, seguida por uma análise de Chatbots e bibliotecas de código livre disponíveis para desenvolvimento do sistema proposto. A definição da arquitetura e tecnologias envolvidas em cada módulo do sistema. Posteriormente, foi realizado o desenvolvimento e implementação de cada módulo do sistema, seguido por uma validação do Chatbot através da interação e *feedback* de usuários. Por fim, foi realizada a análise dos resultados alcançados através das avaliações dos usuários.

3. Fundamentação Teórica

Os termos “chatbot”, “chatterbot” ou simplesmente “bot”, podem ser definidos como um programa de computador com o objetivo de tentar simular um ser humano em uma conversa com um usuário humano. Para que essa conversa ocorra, a forma de interação é através da linguagem natural que, ao contrário da linguagem formal não pode ser diretamente traduzida em operações matemáticas precisas, mas pode ter informações e instruções extraídas, processadas e respondidas na mesma linguagem, sendo esse mecanismo de diálogo denominado de *chatbot* (HAPKE; LANE; HOWARD, 2019).

A atual popularidade dos Chatbots vem em conjunto com a maior quantidade de informações disponíveis na internet, o que acaba necessitando uma forma mais eficiente de acessá-la. Por exemplo, tradicionalmente, para que um usuário tenha uma dúvida sanada em um determinado sistema *web*, era necessário enviar um e-mail ao contato do site ou buscar a dúvida em uma lista de perguntas e respostas. Com utilização de Chatbots, essas informações passaram a ser entregues de forma mais rápida, necessitando apenas a digitação do assunto ou pergunta em um campo de texto. Isso acaba facilitando o acesso e agilizando o processo de busca por informação.

Atualmente, uma das principais barreiras de desenvolvimento de Chatbots é o processamento da linguagem natural (PLN), ou seja, como extrair a informação de uma sentença fornecida pelo usuário de forma eficaz e coerente. O PLN é um campo da ciência que estuda a interação entre computadores e a linguagem humana (ou linguagem natural). Mais precisamente, como os computadores podem extrair, processar e analisar dados de linguagem natural.

Um dos maiores desafios dos Chatbots é entender o sentido de uma sentença em

linguagem natural, isso porque a linguagem humana é altamente ambígua e variável. Goldberg (2017) cita como exemplo as seguintes frases: "Eu como pizza com amigos" e "Eu como pizza com azeitonas". As frases são ambíguas, pois alterando apenas a palavra azeitona para amigos, o sentido completo da frase é alterado. Além disso, é altamente variável, pois a mesma mensagem passada pela frase "Eu como pizza com amigos" poderia ser passa com a frase "Eu divido pizza com parceiros". Outra característica importante é que a linguagem natural está em constante mudança e evolução (GOLDBERG, 2017).

As características das linguagens naturais citadas acima proporcionam um desafio para a construção de algoritmos e métodos computacionais capazes de interpretar o principal sentido de uma frase em meio à um diálogo. Contudo, tanto o desafio do entendimento quanto da produção de linguagem natural utilizando-se computadores pode ser enfrentado através de métodos e algoritmos de aprendizado de máquina que processam dados de linguagem (GOLDBERG, 2017).

A distância de Levenshtein é um método para calcular a diferença entre duas sequências de texto, esta técnica foi desenvolvida pelo matemático soviético Vladimir Levenshtein em 1965 (KASHEFI; SHARIFI; MINAIE, 2013; HOSSAIN et al., 2019). Este método é aplicado em diversas áreas onde existe a necessidade de comparação entre sequências de caracteres, sendo utilizado desde a comparação de sequências de DNA até corretores ortográficos (SANDERS; CHIN, 2009; HOSSAIN et al., 2019). No método de Levenshtein, quanto maior a diferença entre as sequências de caracteres comparados, maior será a distância de Levenshtein, logo, para duas palavras idênticas, a distância de Levenshtein é zero. A similaridade é calculada com base no número mínimo de operações requeridas para transformar uma palavra em outra, sendo que as operações disponíveis são a inserção, deleção e substituição de caracteres. Devido a capacidade de calcular a similaridade entre palavras, a distância de Levenshtein acaba sendo uma técnica muito utilizada por aplicações do tipo Chatbot (KOWSHER et al., 2019; RUF et al., 2020; MASLOWSKI; LAGARDE; CLAVEL, 2017).

Existem diversas bibliotecas para processamento de linguagem natural, como *spaCy* (HONNIBAL; MONTANI, 2021), *NLTK* (em inglês, *Natural Language Toolkit*) (BIRD EDWARD LOPER, 2021), *TextBlob* (LORIA, 2021), *Flair* (AKBIK; BLYTHE; VOLLGRAF, 2018), entre outras. A biblioteca *spaCy*, destaca-se por ser uma biblioteca de código aberto escrita em Python, que suporta mais de 64 linguagens, incluindo a língua portuguesa. O processamento é realizado através da quebra de textos em unidades menores, um processo chamado de *tokenização*. Como parte do processamento, a biblioteca também cria rótulos para cada unidade, denominando as unidades como substantivos, verbos, adjetivos, advérbios, etc, além de criar relações de dependências através da categorização de sujeitos e objetos. Além disso, a biblioteca também é capaz de deflexionar palavras para determinar o seu lema, por exemplo, o lema das palavras "visitarmos", "visitamos", "visitou" é "visitar". Por fim, a biblioteca também é capaz de verificar similaridade entre textos e classificá-los com rótulos e categorias. Essas funcionalidades da biblioteca *spaCy* devem-se principalmente à diversos algoritmos de aprendizado de máquina em sua composição. Portanto, devido às essas características, esta biblioteca é muito utilizada na construção de Chatbot (FINCH et al., 2020; MOPPEL, 2018).

O aprendizado de máquina (em inglês, *machine learning*) é o mecanismo pelo qual busca-se com que uma máquina tenha a capacidade de aprender sem programá-la

explicitamente para isso (SINGH, 2018). No contexto de Chatbot, o aprendizado de máquina se aplica quando o sistema é treinado com conjuntos de perguntas e respostas e posteriormente testado. Após o treinamento, o Chatbot deve ser capaz de responder não somente perguntas exatamente idênticas às treinadas, mas também perguntas similares que apresentam o mesmo sentido. Atualmente existem diversas ferramentas de código aberto desenvolvidas para atuar como mecanismos (ou *engines*, em inglês) para sistemas de diálogo. A biblioteca Python ChatterBot (COX, 2021) é uma coleção de recursos que utilizam aprendizado de máquina e servem para esse propósito, sendo elegível para o desenvolvimento do presente trabalho.

4. Desenvolvimento

4.1. Arquitetura do Sistema

O sistema Chatbot foi dividido em três serviços (ou módulos) denominados: Chatbot API, Chatbot Admin e Chatbot Frontend. Além dos serviços do sistema, a arquitetura também é composta por um banco de dados relacional, o qual persistirá todos os dados pertinentes ao sistema, incluindo a persistência requerida pela biblioteca Python ChatterBot. A Figura 1 mostra uma visão geral de cada módulo e uma descrição de suas principais atribuições. As seções abaixo descrevem cada um dos serviços que compõe o sistema, juntamente com as tecnologias envolvidas.

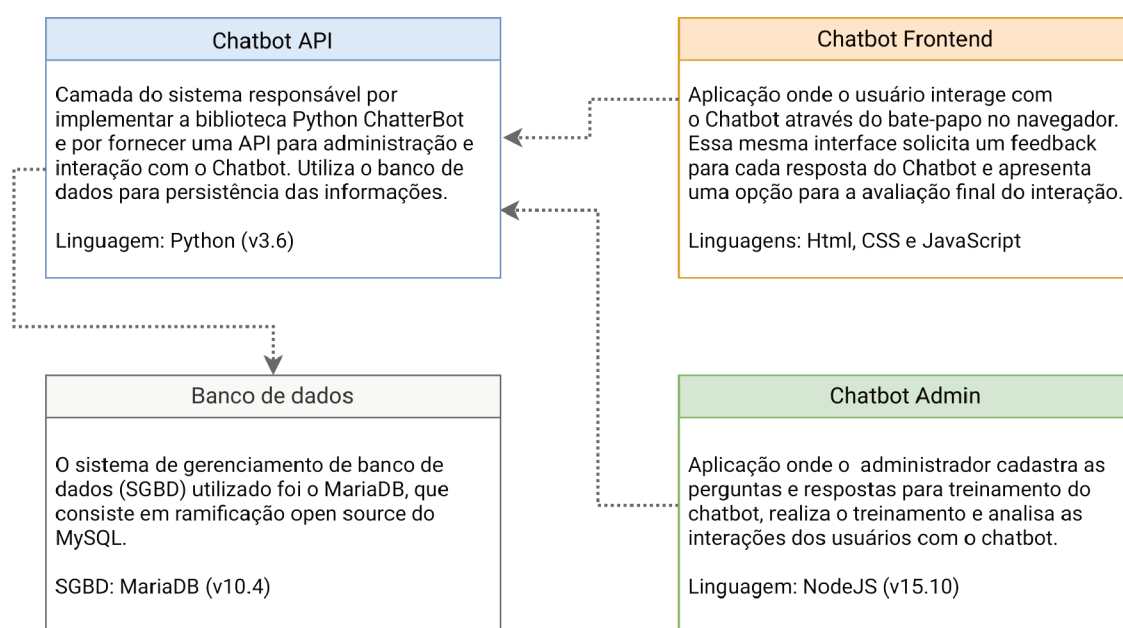


Figure 1. Serviços que compõe o sistema proposto no trabalho.

4.1.1. Serviço Chatbot API

A API do Chatbot foi desenvolvida utilizando a linguagem Python e o *framework Flask* (RONACHER, 2021). A escolha da linguagem foi realizada devido à existência de uma biblioteca de código aberto, denominada ChatterBot. Essa biblioteca foi desenvolvida

para construção de respostas automatizadas em linguagem natural (COX, 2021), para isso, a ferramenta utiliza uma seleção de algoritmos de aprendizado de máquina para produzir diferentes tipos de respostas. O treinamento executado na biblioteca Python ChatterBot pode ser realizado através da inserção de pares de perguntas e respostas. O mecanismo utilizado está baseado na utilização de outra biblioteca de PLN, spaCy. Desta forma através da spaCY, a biblioteca Python ChatterBot é capaz de armazenar as sentenças com extração do lema das palavras fornecidas, geração de *tokens* e remoção de *stop words*, que são as palavras que tem baixa influência no sentido da frase, como: "o", "a", "é", entre outras.

4.1.2. Serviço Chatbot Frontend

O módulo Chatbot Frontend consiste na interface responsável pela interação do usuário com o Chatbot. A interação é realizada através de uma página acessada pelo navegador de internet, onde é fornecido um formulário para interação com o *chatbot* via texto.

As tecnologias envolvidas no desenvolvimento desse serviço consistem em HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*) e JavaScript. Optou-se por essas tecnologias por serem interpretadas no navegador do usuário, sem a necessidade de um processamento no servidor, além do fato de serem tecnologias simples e amplamente utilizadas para criação desse tipo de interface. Para auxiliar no desenvolvimento optou-se também pela utilização de uma biblioteca de funções JavaScript chamada jQuery.

4.1.3. Serviço Chatbot Admin

Para o desenvolvimento do serviço de administração optou-se pela linguagem Node.js (DAHL, 2021). Uma vez que esse sistema necessita comunicar-se com o serviço Chatbot API e também fornecer uma interface para o usuário administrador realizar os procedimentos com boa usabilidade, a linguagem Node.js apresenta-se como uma boa alternativa código aberto, possuindo bibliotecas que facilitam o desenvolvimento de sistemas que necessitam de um processamento no servidor e também um fornecimento de arquivos estáticos (interface de usuário).

4.1.4. Persistência de dados

O armazenamento dos dados do *chatbot* é realizado com a utilização do Sistema de Gerenciamento de Banco de Dados (SGBD) MariaDB na versão 10.4.17 (WIDENIUS, 2021). O banco de dados MariaDB é uma ramificação do projeto MySQL após sua aquisição pela Oracle Corporation, dessa forma, o projeto MariaDB permanece gratuito e de código aberto. Trata-se de um banco de dados do tipo relacional, que possui uma fácil integração com a linguagem Python, sendo também compatível com a biblioteca Python ChatterBot.

4.2. Modelagem

4.2.1. Diagrama de Atividades

A Figura 2 mostra a sequência de ações que ocorrem na interação entre usuário e Chatbot. Inicialmente o usuário acessa a página *web* do Chatbot, ao clicar no ícone referente à abertura do bate-papo, o usuário tem a possibilidade de escrever uma mensagem de texto no campo da caixa de bate-papo. Em seguida, a aplicação retorna uma resposta para o usuário através de uma mensagem na mesma caixa de bate-papo. Para que o usuário possa escrever outra mensagem, é obrigatória a avaliação da mensagem retornada pelo Chatbot, selecionando um ícone com valor positivo ou negativo. O usuário pode continuar a interagir com o Chatbot, avaliando cada resposta, ou pode selecionar a opção "Encerrar e Avaliar", a qual abrirá uma segmento da tela para a avaliação da interação através da escolha de um valor de 1 a 5.

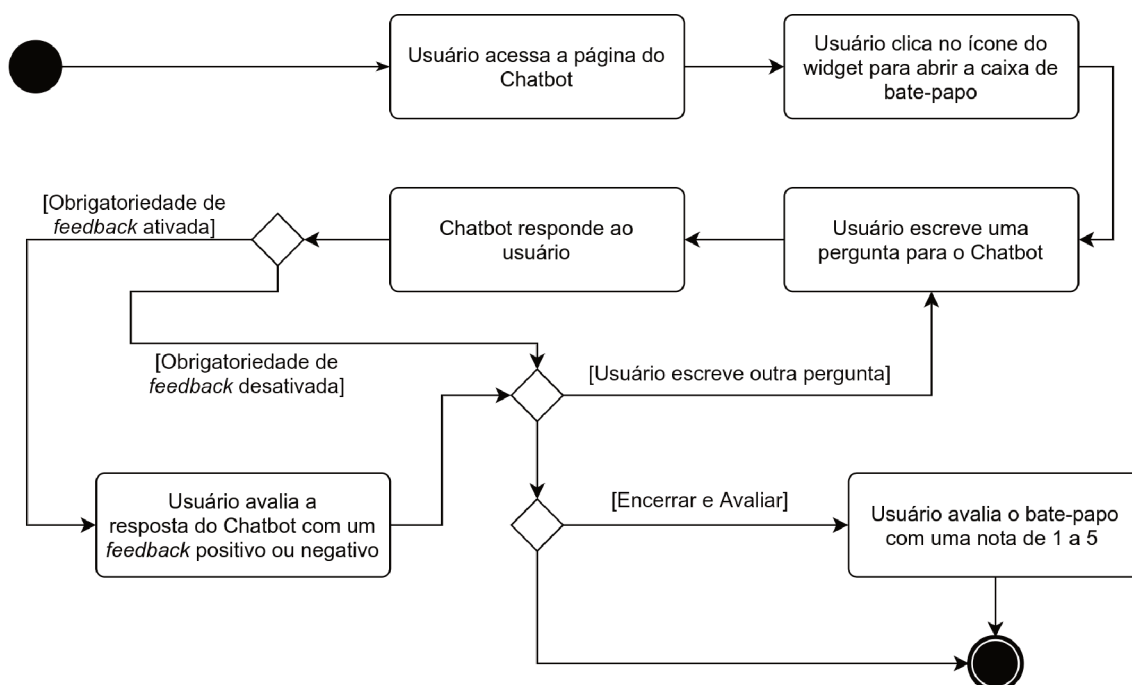


Figure 2. Diagrama de atividade UML relacionado à interação do usuário com o Chatbot.

A Figura 3 apresenta o diagrama de atividades que representa a sequência de ações para realizar o aumento do conhecimento do Chatbot e seu treinamento. Para realizar essa tarefa é necessário acessar a página *web* da área administrativa, efetuar o *login* através do usuário e senha. Uma vez conectado à área administrativa, o administrador deve acessar o menu "Treinamento". Na tela de treinamento, o administrador pode observar os pares de pergunta e resposta previamente cadastrados e optar por treinar o Chatbot ou cadastrar novas informações. Ao optar por cadastrar novas informações, o administrador será direcionado a um formulário para cadastro de pergunta e resposta. Nessa etapa, o administrador pode inserir variações de perguntas para uma única resposta. Após o cadastro, o administrador voltará a tela de treinamento e poderá optar por treinar o Chatbot, incorporando assim as novas informações no banco de dados da biblioteca.

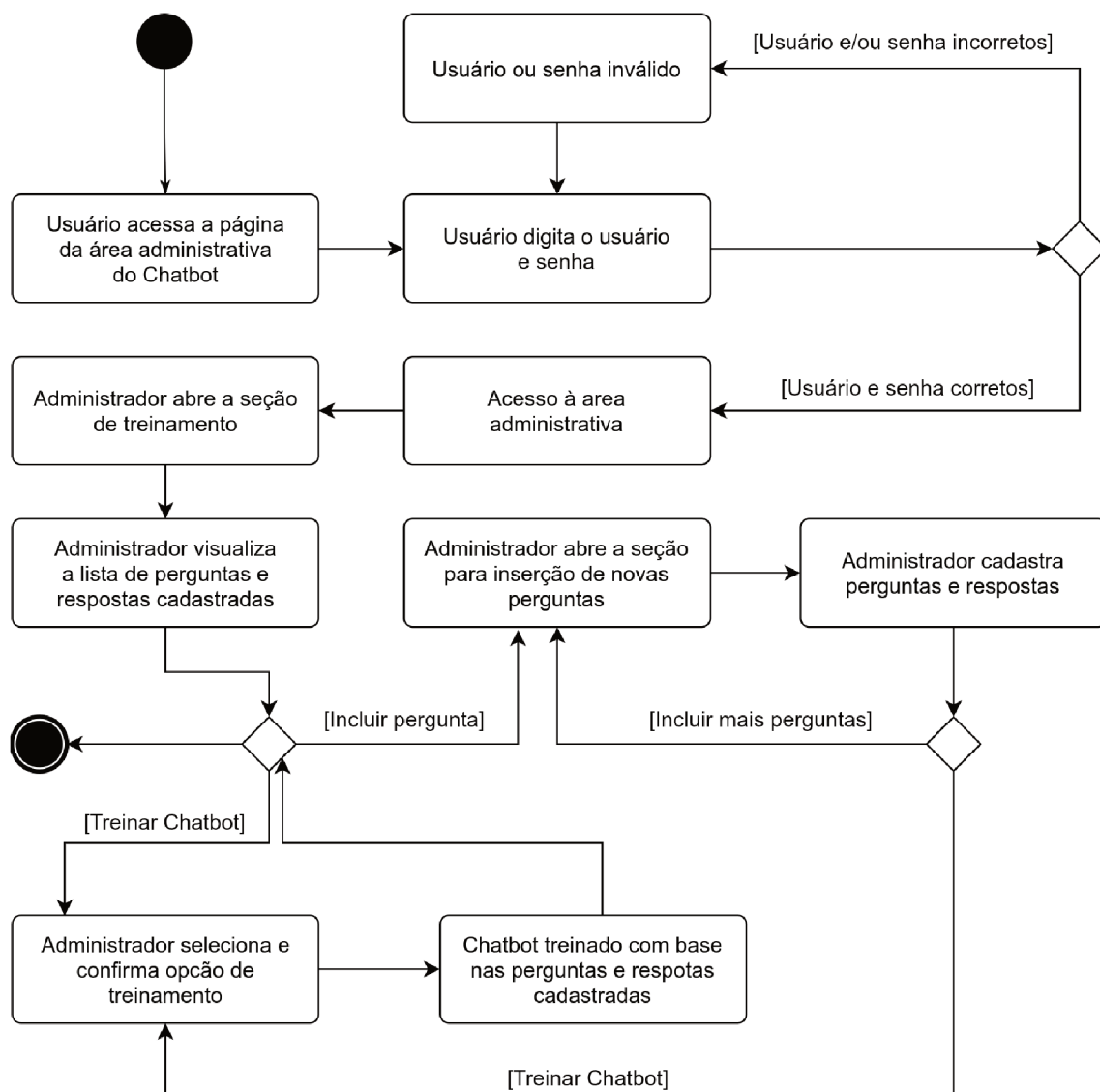


Figure 3. Diagrama de atividade UML relacionado à ação de treinamento realizada pelo administrador.

4.3. Implementação

Cada módulo do sistema atua como um serviço *web* independente, ou seja, eles podem ser executados em diferentes locais. Os módulos utilizam o padrão arquitetural REST (*Representational State Transfer*), dessa forma comunicam-se entre eles através dos métodos HTTP. As solicitações e respostas realizadas entre os módulos são realizadas utilizando-se o formato JSON (*JavaScript Object Notation*). Desta forma, cada serviço pertencente ao sistema foi desenvolvido separadamente.

4.3.1. Serviço Chatbot Frontend

Este módulo consiste na página de bate-papo onde o usuário interage com o Chatbot. A Figura 4 mostra o resultado do desenvolvimento desta interface. Na página, o usuário

necessita clicar no *widget* "Posso ajudar?" do canto inferior direito para que a caixa de bate-papo seja visualizada na tela. Ao digitar uma mensagem na caixa de bate-papo, o usuário receberá uma mensagem de retorno do Chatbot. Neste momento a caixa de envio de mensagem ficará inativa até que o usuário realize o *feedback* da resposta do Chatbot, indicando se a respostas foi útil através do ícone 👍, ou se a resposta não foi útil através do ícone 👎. Essa obrigatoriedade de *feedback* pode ser desativada através de uma variável de ambiente do serviço Chatbot Frontend.

Ao final da interação com o Chatbot, o usuário tem a opção de realizar uma avaliação final, para isso o botão "Encerrar e avaliar" deve ser pressionado (Figuras 4). Esse evento abre uma caixa de avaliação na parte superior do bate-papo, possibilitando o usuário selecionar o número de estrelas (ícone ★) que representa a utilidade geral das respostas recebidas pelo Chatbot.



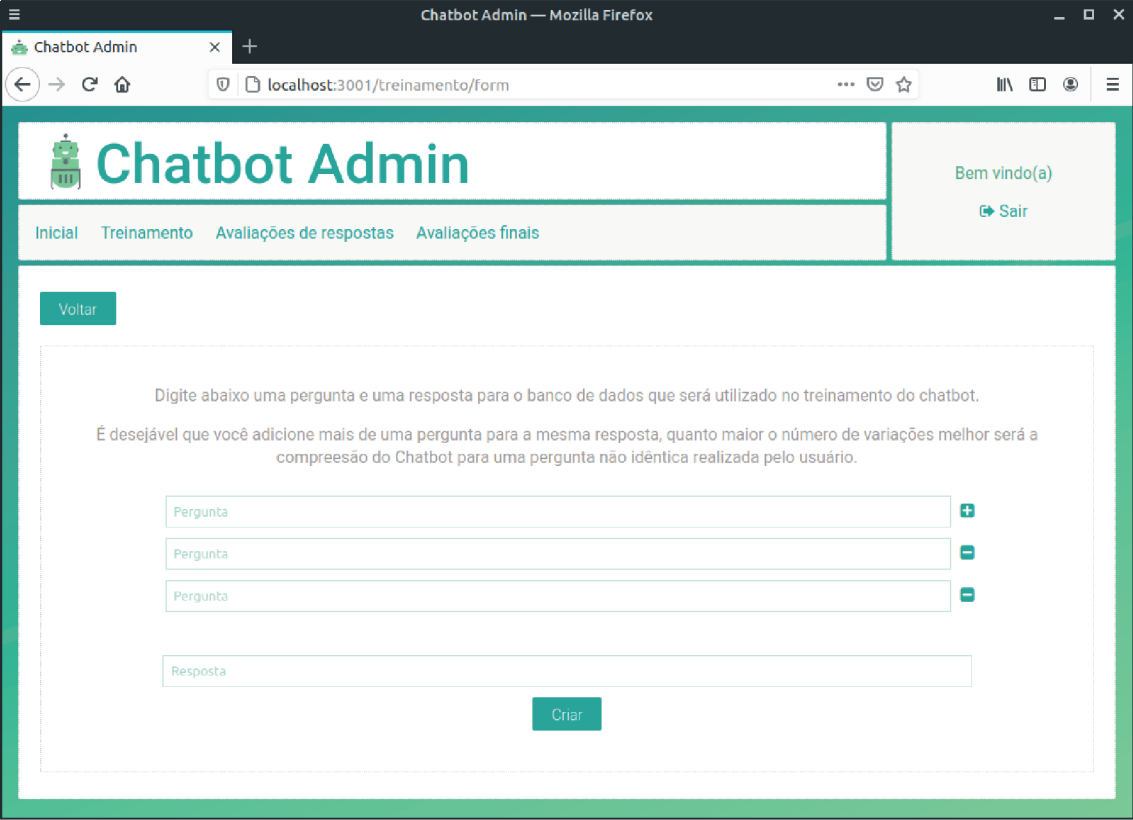
Figure 4. Interface *web* do Chatbot com a caixa de avaliação final aberta.

4.3.2. Serviço Chatbot Admin

O acesso à área administrativa necessita de autenticação através de um usuário e senha, mantendo uma sessão ativa por um determinado tempo. Para manter essa sessão ativa foi utilizada a biblioteca cookie-session, a qual mantém os dados da sessão codificados e armazenados no cookie do usuário autenticado. Após a autenticação com usuário e senha válidos, o usuário administrador é direcionado à página principal da área administrativa.

Na seção "Treinamento", o administrador tem acesso às perguntas e respostas cadastradas. Ao selecionar a opção "Treinar Chatbot", o sistema irá utilizar todas as perguntas e respostas cadastradas nesta seção para treinar a biblioteca ChatterBot. Uma mensagem informando o término do treinamento é mostrado ao usuário. Assim, o administrador pode cadastrar novas perguntas e respostas acionando o botão "Criar nova pergunta". Uma vez criada a pergunta e resposta, é necessário treinar novamente o Chatbot para que essas novas informações sejam adicionadas à base de conhecimento do Chatbot.

O cadastro de novas perguntas e resposta é realizado por meio de um formulário (Figura 5). Neste formulário é possível adicionar mais de uma pergunta para a mesma resposta. Isto é importante para aumentar a assertividade do Chatbot. Uma vez que a biblioteca é capaz de reconhecer sentenças semelhantes, com um grau de similaridade ajustável, ao adicionar variações de perguntas que possam ser respondidas pela mesma resposta, a assertividade do Chatbot é aumentada.



The screenshot shows a web browser window titled "Chatbot Admin" with the URL "localhost:3001/treinamento/form". The page has a green header with the "Chatbot Admin" logo and navigation links: "Inicial", "Treinamento", "Avaliações de respostas", and "Avaliações finais". A "Voltar" button is in the top left. The main content area contains a form with the following elements:

- A "Voltar" button in a green box.
- Text: "Digite abaixo uma pergunta e uma resposta para o banco de dados que será utilizado no treinamento do chatbot."
- Text: "É desejável que você adicione mais de uma pergunta para a mesma resposta, quanto maior o número de variações melhor será a compreensão do Chatbot para uma pergunta não idêntica realizada pelo usuário."
- Three "Pergunta" input fields, each with a "+" icon to its right and a "-" icon to its left.
- A "Resposta" input field.
- A "Criar" button in a green box at the bottom center.

Figure 5. Interface web - cadastro de novos conhecimentos através de pergunta(s) e resposta.

Com o objetivo de avaliar as interações entre o Chatbot e os usuários, a seção “Avaliações de respostas” mostra registro de *feedback* para um determinado par de pergunta e resposta (Figura 6). Dessa forma, é possível avaliar os *feedbacks* com base na pergunta e resposta, auxiliando o administrador no cadastro de novos conhecimentos para o Chatbot.

A página de avaliações finais mostra a avaliação registrada por um usuário em determinada sessão. Em cada acesso ao serviço Chatbot Frontend, um id de sessão único é gerado, caso o usuário recarregue a página, um novo id é gerado. Este id de sessão é enviado sempre que um *feedback* ou uma avaliação final sejam enviados pelo usuário. Isto possibilita o administrador buscar as informações de determinada avaliação na seção de respostas. Ou seja, se o administrador identificar uma avaliação de apenas uma estrela (muito insatisfeito), ele pode copiar esse id e buscar todas as perguntas e respostas na aba “Avaliação de respostas”, isso o ajudará a criar perguntas mais assertivas para o conhecimento do Chatbot com base nessa interação não satisfatória.

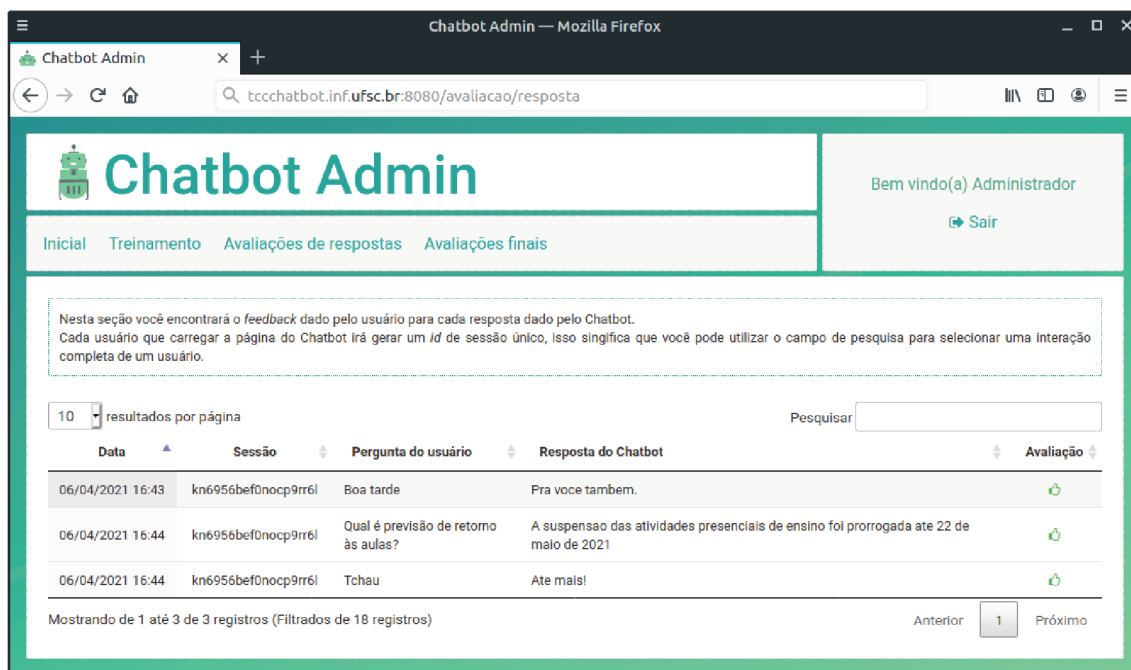


Figure 6. Área de visualização de *feedbacks* para as respostas fornecidas pelo Chatbot.

As funcionalidades das seções “Avaliações de respostas” e “Avaliações finais” são semelhantes às descritas para a seção de “Treinamento”, com a capacidade de paginação, busca, ordenação e seleção de resultados por página.

5. Resultados

O sistema foi implementado e disponibilizado através de um endereço eletrônico com o objetivo de ser avaliado por alunos do curso de Sistemas de Informação da UFSC. A disseminação do endereço do Chatbot foi realizada através do envio de um e-mail pela coordenadoria do curso, no qual, usuários voluntariamente interagiram com o Chatbot. Os resultados discutidos nesta seção representam interações que ocorreram em um período de seis dias de exposição do sistema.

O número total de mensagens enviadas ao Chatbot foi de 394. Deste total, 271 foram consideradas perguntas válidas e 123 foram consideradas inválidas. O conjunto considerado inválido foi representado por mensagens que não representam uma pergunta propriamente dita, tentativas de invasão ao sistema, tentativas de inativação do sistema, mensagens ofensivas e mensagens com erros ortográficos e gramaticais.

Em relação aos *feedbacks* realizados pelos usuários a cada respostas do Chatbot, obteve-se um número de 140 (51,7%) para avaliações positivas e 131 (48,3%) para avaliações negativas. Ao avaliar individualmente os *feedbacks* negativos, observou-se que muitas perguntas direcionadas ao Chatbot não possuem respostas por não se tratarem do contexto de respostas do Chatbot. Uma vez que o objetivo da aplicação é auxiliar sobre questões pertinentes ao curso de Sistemas de Informação da UFSC.

O total de sessões abertas no navegador que interagiram com o Chatbot foi 83. Dentre este total, apenas 33 realizaram a avaliação final da interação. Isto significa que

nem todos os usuários que interagiram com o Chatbot realizaram a avaliação final. A Figura 7 mostra o resultado do julgamento dos usuário que realizaram a avaliação selecionando uma das opções de valor, de 1 estrela (muito insatisfeito) até 5 estrelas (muito satisfeito).

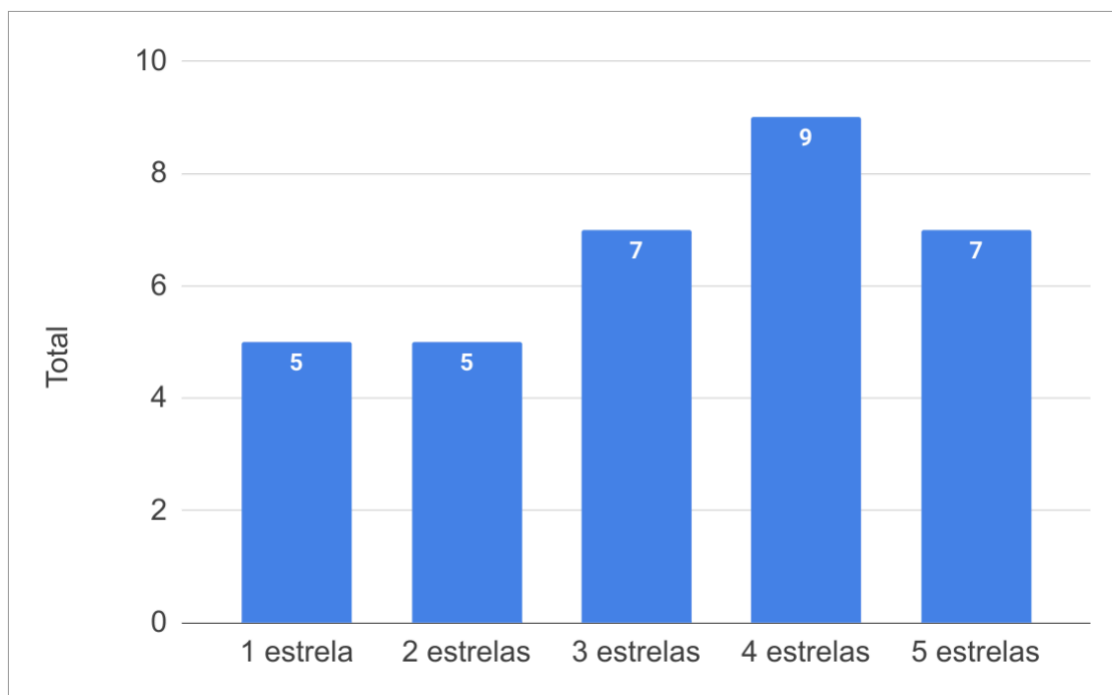


Figure 7. Avaliações finais registradas pelos usuários.

Saudações, agradecimentos, despedidas e questionamentos fora do contexto de Sistemas de Informação foram os assuntos mais evidentes nas interações dos usuários. O segundo assunto mais evidenciado foi questionamentos sobre o Chatbot, por exemplo, nome, idade, humor, funcionalidades, etc. Em seguida observou-se assuntos pertinentes ao curso, como questionamentos sobre a matrícula, contatos da coordenação e do coordenador, bem como a duração do curso, informações sobre o site, grade de horários, estágio, localização física e trancamento de matrícula. Com base no número de ocorrências de perguntas em determinados assuntos é possível, através do painel administrativo, cadastrar novos conhecimentos utilizando esta informação como um guia para aumentar a assertividade do Chatbot melhorando assim a utilidade da ferramenta.

5.1. Conclusão

O presente trabalho atingiu o objetivo geral criando um sistema do tipo Chatbot com base em uma biblioteca de código aberto. Sobre essa biblioteca foram desenvolvidas camadas do sistema que resultaram em uma aplicação capaz de interagir com um usuário no formato de bate-papo, utilizando a linguagem natural através de uma interface *web*. Além de uma área administrativa para visualizar as interações e, com base nessas informações, criar perguntas e respostas para treinamento do sistema aumentando sua base de conhecimento.

A interface *web* de bate-papo, serviço “Chatbot Frontend”, mostrou-se suficiente para permitir a interação dos usuários através de linguagem natural em formato de men-

sagem de texto, uma vez que não foi encontrada nenhuma dificuldade por parte dos usuários no envio de mensagens para o Chatbot. Assim como a área administrativa, “Serviço Admin”, mostrou-se eficiente para administrar o Chatbot, através das funcionalidades básicas de Treinamento e Avaliações. No entanto, sugere-se com trabalho futuro a validação desta área juntamente com a coordenadoria e secretaria do curso de Sistemas de Informação.

O levantamento de dados, descrito na seção de resultados, sobre os principais assuntos abordados pelos usuários, fornece uma base para auxiliar o administrador no cadastro de novas perguntas e respostas, de forma direcionada aos assuntos mais solicitados. Juntamente com o gerenciamento das informações observadas nos menus de “Avaliações finais” e “Avaliações de respostas” do painel administrativo, fornecem uma base para direcionar o usuário administrador sobre quais as perguntas são pertinentes aos alunos do curso, evidenciando também quais não são contempladas pelo conhecimento atual do Chatbot, necessitando ser incorporadas na base de dados. Desta forma, o acréscimo da base de conhecimento proporcionará uma melhora na qualidade das respostas tornando a ferramenta cada vez mais útil na busca por informações sobre o curso de Sistemas de Informação, tanto para alunos do curso como para pessoas externas à ele, que buscam alguma informação de forma ágil.

O resultado das avaliações finais mostra que as interações obtiveram resultados variados de satisfação, de muito insatisfeito até muito satisfeito. Isto mostra a dificuldade em se trabalhar com um sistema que interage com usuários sem regras claras de comunicação. A linguagem utilizada pelos usuários compreendeu desde palavras abreviadas, gírias até a descontextualização do objetivo da interação. Além disso, o número total de perguntas e respostas utilizado no treinamento do Chatbot para essa primeira exposição a usuários reais foi de 150. Este é um valor ainda pequeno para um Chatbot poder fornecer uma variedade de respostas e abranger todo o domínio de conhecimento pertinente às questões do curso de Sistemas de Informação. Levando-se em conta que foi realizado apenas um ciclo de interação com usuários reais, a base de conhecimento do Chatbot pode evoluir para uma assertividade maior. Uma vez que, as mensagens coletadas no primeiro ciclo servem de matéria-prima para o cadastro e treinamento de novas perguntas e respostas. Portanto, o gerenciamento da base de conhecimento deve ser incremental com o intuito de aumentar em relação ao tamanho e assertividade.

Referências

AKBIK, A.; BLYTHE, D.; VOLLGRAF, R. Contextual string embeddings for sequence labeling. In: *COLING 2018, 27th International Conference on Computational Linguistics*. [S.l.: s.n.], 2018. p. 1638–1649.

BIRD EDWARD LOPER, E. K. S. *NLTK*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://nltk.org>>.

COX, G. *ChatterBot: Machine learning in Python*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://chatterbot.readthedocs.io>>.

DAHL, R. *Node.js*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://flask.palletsprojects.com>>.

- FINCH, S. E. et al. Emora: An inquisitive social chatbot who cares for you. *arXiv preprint arXiv:2009.04617*, 2020.
- GOLDBERG, Y. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, v. 10, n. 1, p. 1–309, 2017.
- HAPKE, H. M.; LANE, H.; HOWARD, C. *Natural language processing in action*. [S.l.]: Manning, 2019.
- HONNIBAL, M.; MONTANI, I. *spaCy*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <https://spacy.io>.
- HOSSAIN, M. M. et al. Auto-correction of english to bengali transliteration system using levenshtein distance. In: IEEE. *2019 7th International Conference on Smart Computing & Communications (ICSCC)*. [S.l.], 2019. p. 1–5.
- KASHEFI, O.; SHARIFI, M.; MINAIE, B. A novel string distance metric for ranking persian respelling suggestions. *Natural Language Engineering*, Cambridge University Press, v. 19, n. 2, p. 259, 2013.
- KOWSHER, M. et al. Doly: Bengali chatbot for bengali education. In: IEEE. *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. [S.l.], 2019. p. 1–6.
- LORIA, S. *TextBlob*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <https://textblob.readthedocs.io>.
- MASLOWSKI, I.; LAGARDE, D.; CLAVEL, C. In-the-wild chatbot corpus: from opinion analysis to interaction problem detection. In: *International conference on natural language and speech processing*. [S.l.: s.n.], 2017.
- MOPPEL, J. *Socratic chatbot*. [S.l.]: University Of Tartu, Institute of Computer Science, Bachelor's Thesis, 2018.
- RAHANE, W. et al. Artificial intelligence based solarbot. In: IEEE. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. [S.l.], 2018. p. 601–605.
- RAHMAN, A.; MAMUN, A. A.; ISLAM, A. Programming challenges of chatbot: Current and future prospective. In: IEEE. *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. [S.l.], 2017. p. 75–78.
- RONACHER, A. *Flask*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <https://flask.palletsprojects.com>.
- RUF, B. et al. Pharmabroad: a companion chatbot for identifying pharmaceutical products when traveling abroad. In: *Information and Communication Technologies in Tourism 2020*. [S.l.]: Springer, 2020. p. 218–228.
- RUSSEL, S.; NORVIG, P. Inteligência artificial. 3a edição. *Editora Campus*, 2013.
- SANDERS, N. C.; CHIN, S. B. Phonological distance measures. *Journal of quantitative linguistics*, Taylor & Francis, v. 16, n. 1, p. 96–114, 2009.
- SINGH, P. *Machine Learning with PySpark: With Natural Language Processing and Recommender Systems*. [S.l.]: Apress, 2018.

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

WIDENIUS, U. M. *MariaDB Foundation*. 2021. Acesso em: 13 de abr. de 2021. Disponível em: <<https://mariadb.org>>.