

# IFBot – Um *Chatbot* como ferramenta de apoio para o acesso a informações do Instituto Federal da Bahia – Campus Salvador

Arla Figueiredo  
Análise e Desenvolvimento de Sistemas  
Instituto Federal da Bahia  
Salvador, Brasil  
Email: arla.figueiredo@gmail.com

Renato Novais  
Departamento de Computação  
Instituto Federal da Bahia  
Salvador, Brasil  
Email: renato@ifba.edu.br

*Abstract— The large volume of data circulating on the internet and the increasing connectivity of individuals in the digital era have been driving the use of automated conversational agents, also known as chatbots. This trend has emerged to meet users' demand for speed and convenience in obtaining information. Brazil has been standing out in this field, being one of the world's leading users of chatbots, with an increase of over 50% in the last two years and a total of 671,000 bots developed. Currently, such systems are employed in various areas and have gained visibility in the academic sector, being widely used in educational institutions for support, dissemination, and facilitating access to specific data. However, even with such a tool already in existence, many Federal Institutes of Education, such as Federal Institute of Bahia - Campus Salvador, do not have an efficient way to provide academic information to their students. Thus, the proposal to develop a chatbot for the same, named IFBot, arises. IFBot aims to simplify the routine of students in searching for academic information, using Natural Language Processing and Machine Learning to enhance the efficiency of question recognition, in addition to having a management platform to keep its content accurate and up-to-date. To evaluate the tool, tests were conducted with IFBA students to obtain a first impression of the system's usefulness. The results obtained were very positive, indicating that the system has great potential.*

*Keywords—Chatbot, Artificial Intelligence, Natural Language Processing*

*Resumo—O grande volume de dados que circula na internet e a crescente conectividade dos indivíduos na era digital têm impulsionado cada vez mais o uso de agentes conversacionais automatizados, também conhecidos como *chatbots*. Essa tendência surgiu para atender a demanda dos usuários por agilidade e conveniência para obter informações. O Brasil vem se destacando nesse campo, sendo um dos principais utilizadores de *chatbots* do mundo, com um aumento de mais de 50% nos últimos dois anos e um total de 671 mil *bots* desenvolvidos. Atualmente tais sistemas são empregados nas mais diversas áreas e têm ganhado visibilidade na área acadêmica, sendo muito utilizado em instituições educacionais para atendimento, divulgação e facilitação do acesso a dados específicos. No entanto, mesmo já existindo tal ferramenta, muitos Institutos Federais de Ensino, como o Instituto Federal da Bahia - Campus Salvador, não contam com uma forma eficiente de disponibilizar informações do meio acadêmico para seus alunos. Neste contexto, o objetivo deste trabalho foi desenvolver um *chatbot* para o mesmo, denominado IFBot. O IFBot simplifica a rotina dos discentes na busca por informações acadêmicas, empregando o Processamento de Linguagem Natural e Aprendizado de Máquina para aumentar a eficiência do reconhecimento das perguntas, além de contar com*

uma plataforma de gestão para manter seu conteúdo correto e atualizado. Para avaliar a ferramenta, foram realizados testes com estudantes do IFBA, a fim de obter-se uma primeira impressão sobre a utilidade e eficácia do sistema. Os resultados obtidos foram muito positivos, indicando que o sistema tem muito potencial.

**Palavras-Chave:** *Chatbot*, Inteligência Artificial, Processamento de Linguagem Natural.

## I. INTRODUÇÃO

A era digital, também conhecida como era da informação, mudou expressivamente a forma como a população se comunica e obtém conhecimento. Observa-se que a cada dia que passa a quantidade de dados cresce exponencialmente. Segundo a IBM, diariamente, são gerados 2,5 quintilhões de dados no mundo, sendo que cerca de 90% destes foram gerados nos últimos 3 anos [1]. Assim, devido a esse vasto volume de informações, a busca por aquelas que são confiáveis e relevantes torna-se uma tarefa cada vez mais difícil e demorada. Paralelamente a isso, os indivíduos também se tornam progressivamente mais conectados e, conseqüentemente, por vivenciarem rotinas agitadas, ficam mais exigentes por rapidez e comodidade para obter informações [2]. É neste contexto que ganham destaque a Inteligência Artificial e duas de suas subáreas, o Processamento de Linguagem Natural (PLN) e o Aprendizado de Máquina.

A partir dos conceitos de tais campos, surgem os agentes de conversação automatizados, também conhecidos como *chatbots*, como soluções dinâmicas para suprir essa crescente demanda por produtividade e conveniência tão almejada pela acelerada sociedade moderna [2]. Embora esta seja uma área de pesquisa relativamente recente, suas aplicações práticas aumentaram substancialmente nos últimos anos, sendo que de acordo com o Mapa do Ecossistema Brasileiro de *Bots* 2023, o Brasil é considerado um dos maiores países na utilização de *chatbots*. No período entre 2022 e 2023, houve crescimento de mais de 50% na criação de *chatbots* brasileiros e a marca atual de *bots* desenvolvidos já atinge o montante de 671 mil [3].

A origem da palavra *chatbot* resulta da união dos termos em inglês *chat* (que significa conversa) e *bot* (abreviatura de *robot*, que significa robô). Assim, de acordo com o Dicionário

Léxico da Língua Inglesa, o *chatbot* é definido como um software construído para simular conversas com usuários humanos [4]. O *chatbot* também pode ser definido como um programa de inteligência artificial e um modelo de interação humano-computador (HCI) e possui diferentes nomenclaturas, tais como: Entidades Artificiais de Conversação, Agentes Interativos, *Bots* Inteligentes e Assistentes Virtuais [5]. Atualmente os mesmos são empregados em diversos campos, como educação, negócios, *e-commerce*, entre outros, e seu recurso mais atrativo é relacionado a produtividade, já que reduzem a demanda por atendimento humano e podem atender múltiplos usuários simultaneamente [5].

Diversos trabalhos já foram feitos com o objetivo de avaliar o emprego de *chatbots* em diferentes contextos educacionais, desde como forma de facilitar a divulgação de notícias no ambiente acadêmico, a exemplo do projeto feito no Instituto Federal de Educação, Ciência e Tecnologia de Brasília (IFB) - Campus Brasília [6], assim como projetos realizados no Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo (IFES) que tiveram o intuito de facilitar o acesso às informações específicas da instituição de ensino e de seu processo seletivo [7][8]. Segundo Okonkwo e Ade-Ibijola (2021), os principais benefícios da aplicação de *bots* na educação é a promoção de rápido acesso a dados educacionais, atendimento simultâneo a múltiplos usuários e suporte imediato [9].

Vale destacar que os Institutos Federais de Ensino como um todo possuem um ampla gama de informações que são diariamente publicados e atualizados nos seus diversos portais institucionais online. Esses dados são distribuídos por múltiplas páginas, divididas em várias seções, *links* e até mesmo em documentos anexados às mesmas [7]. Não diferente, este cenário também é uma realidade no Instituto Federal da Bahia (IFBA) - Campus Salvador.

O IFBA - Campus Salvador oferece uma variedade de cursos que englobam diferentes modalidades educacionais, tais como Ensino Técnico (Integrado e Subsequente), PROEJA, Graduação e Pós-Graduação (Especialização, Mestrado, Doutorado). Tal diversidade acadêmica acaba ocasionando uma multiplicidade de informações que dificultam o acesso eficiente e eficaz a dados essenciais. Diferentemente de alguns outros Institutos Federais Brasileiros, o campus Salvador não conta com uma página detalhada de “Dúvidas Frequentes”, de forma que as documentações de regulamentação de ensino, tanto do nível técnico quanto do nível superior, com suas múltiplas páginas e mais de 100 artigos escritos com uma linguagem formal nem sempre é a primeira opção buscada pelos alunos para sanar suas dúvidas. Assim, os discentes têm que recorrer a outros meios para obter as informações desejadas, que geralmente consistem nos múltiplos canais de comunicação (*e-mail*, *whatsapp*, *telegram*, *instagram*) das coordenações e/ou outros setores administrativos, que pela alta demanda não contam com um rápido atendimento.

Considerando os problemas mencionados e com o objetivo de otimizar o dia a dia dos alunos, este trabalho tem como objetivo desenvolver um assistente virtual (*chatbot*) para o IFBA - Campus Salvador, denominado IFBot. O foco é agilizar e simplificar a busca por informações do ambiente acadêmico, como normas, regimentos, calendários acadêmicos, estruturas curriculares, localizações de salas de aula, etc. Através da integração das tecnologias de PLN e Aprendizado de Máquina,

espera-se que o mesmo seja capaz de compreender múltiplas perguntas e fornecer respostas relevantes. Além disso, o IFBot contará com uma plataforma de gestão para que se possa manter seu conteúdo constantemente atualizado e integrado com as demandas dos estudantes. Para avaliar a ferramenta, foram realizados testes com estudantes do IFBA, a fim de obter-se uma primeira impressão sobre a eficácia e utilidade do sistema. Os resultados obtidos foram muito positivos, indicando que o sistema tem muito potencial.

Além desta seção, a estrutura deste trabalho se desdobra da seguinte maneira: a Seção II apresenta a fundamentação teórica, composta por artigos e citações que fundamentaram a elaboração do projeto. A Seção III trata dos trabalhos correlatos a esta solução. A Seção IV apresenta a solução desenvolvida. A Seção V apresenta o estudo conduzido para validar a solução. Por fim, na Seção VI são tecidas as conclusões, abordando tanto as perspectivas futuras do projeto quanto seus impactos esperados.

## II. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas a seguir as bases teóricas sobre as quais o presente trabalho foi projetado e implementado. A subseção II-A apresenta o conceito de Inteligência Artificial e um breve histórico. A subseção II-B apresenta o conceito de Aprendizado de Máquina, seus tipos, abordagens e aplicações. A subseção II-C apresenta o conceito de Processamento de Linguagem Natural, seu histórico, etapas e aplicações. Por fim, A subseção II-D apresenta o conceito *chatbots*, seu histórico e tipos.

### A. Inteligência Artificial

Ao longo de milênios, a humanidade buscou compreender o funcionamento de seu próprio pensamento, ou seja, como um organismo pode adquirir a capacidade de compreender, prever e interagir com um mundo extremamente complexo? Foi a partir desses anseios que surgiu a Inteligência Artificial (IA) [10]. Tal termo foi cunhado por John McCarthy, um professor da Universidade Stanford, em 1955. Ele definiu a IA como “a ciência e engenharia de criar máquinas inteligentes” [11].

No que diz respeito ao conceito, quando se fala em Inteligência Artificial não há um consenso sobre a sua definição. No entanto, segundo Russell e Norvig (2013), ao longo do tempo estabeleceu-se quatro linhas de pensamento [10]:

- I. Sistemas que pensam como seres humanos;
- II. Sistemas que atuam como seres humanos;
- III. Sistemas que pensam racionalmente;
- IV. Sistemas que atuam racionalmente [10].

Gomes (2010) afirma que, historicamente, todas as quatro linhas de pensamento para o estudo da inteligência artificial têm sido seguidas. Existem as abordagens centradas em torno de seres humanos e abordagens centradas em torno da racionalidade, além de haver uma certa rivalidade entre elas [12].

De acordo com Russell e Norvig (2013), por ter se originado após a Segunda Guerra Mundial, entre as décadas de 40 e 50, o campo da Inteligência Artificial é considerado um dos mais recentes na área de Ciências Exatas [10]. A seguir vamos

abordar um breve histórico de suas principais fases para obter-se um panorama mais amplo do seu desenvolvido ao longo dos anos.

O primeiro grande trabalho reconhecido como IA foi idealizado e executado em 1943 por Warren McCulloch e Walter Pitts, no qual os mesmos propuseram um modelo de neurônios artificiais, que seriam os precursores para que futuramente surgisse o Aprendizado de Máquina, conceito abordado na subseção II-B. Porém foi em 1950, por meio de Alan Turing, que a IA ganhou mais notoriedade no campo científico [13].

Alan Turing desenvolveu o chamado “Teste de Turing”, também conhecido como “O Jogo da Imitação” que avalia se uma máquina é capaz de simular a comunicação escrita de um humano. De acordo com Barbosa e Bezerra (2020) “a ideia do experimento era verificar se a máquina poderia emitir informações como se fosse uma pessoa, sem gerar desconfianças no receptor de que se tratava de um programa de computador. Se isso ocorresse e pelo menos um terço dos participantes se sentisse convencido de que o diálogo travado havia sido com um humano, a máquina poderia ser considerada ‘inteligente’ ” [13].

Conforme apontam Barbosa e Bezerra (2020), ainda no mesmo ano em que o experimento com o teste aconteceu, Alan Turing publicou o artigo “*Computing Machinery and Intelligence*” (Máquinas Computacionais e Inteligência) que é considerado o texto precursor da Inteligência Artificial. E assim, por meio das contribuições de Turing, criou-se um grande entusiasmo que alavancou uma série de investimentos na área de IA [13].

A partir daí, foram desenvolvidos diversos sistemas “inteligentes”, como programas capazes de jogar xadrez, planejar tarefas, analisar estruturas moleculares, etc. [14]. Em 1964, surgiu a ELIZA, considerado o primeiro *chatbot* da história, desenvolvida por Joseph Weizenbaum, no laboratório de Inteligência Artificial do MIT (*Massachusetts Institute of Technology*). O objetivo da ELIZA era conversar simulando uma psicanalista, sendo que o programa já possuía respostas prontas e respondia baseando-se na correspondência de palavras-chave [13]. Apesar de hoje tal abordagem ser considerada muito básica, naquela época foi inovador. De acordo com Barbosa e Bezerra (2020), “O robô apresentava aconselhamentos psicológicos e frases sugestivas de empatia, a ponto de muitos o considerarem um possível complemento nos tratamentos psicoterápicos” [13].

No entanto, apesar da empolgação gerada inicialmente, entre as décadas de 70 e 80 ocorreu o chamado “inverno da IA”, pois nesta fase, os pesquisadores começaram a deparar-se com algumas limitações tecnológicas, como problemas relacionados ao armazenamento de dados e ao tempo de processamento [15][14]. Conforme aponta Anyoha (2017), os computadores daquela época não tinham a capacidade de armazenar dados suficientes ou não conseguiam processar os dados com a rapidez necessária, o que impactava diretamente em diversas tarefas, como as de processamento de linguagem, em que é necessário conhecer o significado de diversas palavras e interpretá-las a partir de múltiplas combinações [16]. Desse modo, devido às limitações políticas e econômicas do período pós-guerra, ocorreu um período de redução de investimentos para financiar as pesquisas na área [13].

A partir da década de 80, a Inteligência Artificial voltou a ser uma área de pesquisa muito ativa [14]. Nessa época, Edward Feigenbaum introduziu o conceito de sistemas especialistas, que tinham o objetivo de imitar o processo de tomada de decisão de um especialista humano, ou seja, o sistema aprendia com um especialista em uma determinada área como agir em diversas situações, e com isso pessoas não especialistas na mesma área poderiam receber aconselhamento deste programa. Tais sistemas foram amplamente empregados nas indústrias e propiciaram a criação de outras tecnologias no campo da IA, como a aprendizagem de máquina e o processamento de linguagem natural, que são amplamente utilizados nos dias atuais [16].

Conforme apontam Ludermir (2021) e Escott (2017), a Inteligência Artificial pode ser dividida em três grupos: IA Focada, também conhecida como *Artificial Narrow Intelligence* (ANI); IA Generalizada ou *Artificial General Intelligence* (AGI), e IA Superinteligente ou *Artificial Superintelligence* (ASI) [17][18].

A *Artificial Narrow Intelligence* (Inteligência Artificial Estreita), também conhecida como IA focada ou IA fraca, é o único tipo de inteligência artificial desenvolvida com sucesso até o momento. De acordo com Escott (2017), embora tais sistemas possam parecer inteligentes, elas não são capazes de replicar a inteligência humana, apenas simulam o comportamento humano com base em parâmetros e contextos. Tal IA é projetada para executar tarefas específicas, como por exemplo, reconhecimento facial, reconhecimento de fala/assistentes de voz, dirigir um carro, etc. A IA Focada conquistou um grande progresso na última década, impulsionado pelo avanço dos hardwares e em pesquisas nas áreas de aprendizado de máquina e processamento de linguagem natural, sendo que tais conceitos serão abordados nas seções a seguir [17].

A *Artificial General Intelligence* (Inteligência Artificial Geral), também conhecida como IA profunda ou IA forte, foi projetada para imitar a inteligência e/ou comportamento humano, não só se limitando a realizar tarefas específicas e sim possuindo a capacidade de aprender e aplicar a sua inteligência para resolver qualquer problema, sendo que suas ações são indistinguíveis da de um ser humano. No entanto, conforme aponta Escott (2017), a falta de conhecimento abrangente sobre como se dá o funcionamento do cérebro humano dificulta os avanços nessa área. Já a *Artificial Superintelligence* (Superinteligência Artificial) não se limita apenas a se equiparar à inteligência humana, mas sim a superá-la. Tais sistemas ainda não existem e não se sabe se serão possíveis de serem criados [17].

## B. Aprendizado de Máquina

O mecanismo de aprendizado humano, ou seja, a capacidade inata de adquirir fatos, habilidades e conceitos mais abstratos tem sido objeto de estudos de biólogos, psicólogos, filósofos e até mesmo de pesquisadores de inteligência artificial [19]. Segundo Carbonell et. al (1983) “é muito mais difícil ensinar uma pessoa explicando cada etapa de uma tarefa complexa do que mostrando a essa pessoa como tarefas semelhantes geralmente são realizadas” [19]. Essa abordagem de ensino também é aplicada na ciência do aprendizado de máquina.

Para Silva (2019), *Machine Learning* ou Aprendizado de Máquina é um subcampo da Inteligência Artificial que consiste na construção de sistemas inteligentes a partir de dados [20]. De forma análoga, Monard e Baranauskas (2003) afirmam que o objetivo do Aprendizado de Máquina é o “desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática a partir de exemplos e observações” [21]. Já de acordo com Silva e Vanderlinde (2012), o aprendizado de máquina nada mais é do que um aprendizado por experiência, e à medida que a tarefa é executada, a máquina aprende a melhor maneira de resolver o problema [22].

Diferentemente das técnicas utilizadas nos Sistemas Especialistas citados anteriormente, em que o processo de transmissão de conhecimento era feito explicitamente e diretamente por uma pessoa, com o advento do aprendizado de máquina, a extração do conhecimento humano pode ser feito de uma forma mais automatizada. Isso possibilitou o desenvolvimento de sistemas inteligentes de forma muito mais eficiente [23].

Tal eficiência é possível pois são aplicados algoritmos que aprendem a partir de dados de treinamento específicos para um dado problema de forma interativa. Isso permite que os computadores localizem padrões ocultos e complexos sem serem explicitamente programados para resolver o problema [23].

Existem três tipos principais de Aprendizado de Máquina: Supervisionado, Não Supervisionado e por Reforço [22].

Segundo Monard e Baranauskas (2003), no aprendizado supervisionado é fornecido ao algoritmo de aprendizado um conjunto de exemplos de treinamento ( $X$ ) em que tem-se os rótulos ( $y$ ), ou seja, tem-se a resposta associada a cada exemplo [21]. Pode-se dizer que cada exemplo é uma tupla  $(X, y)$ . Conforme aponta Batista (2003), o principal objetivo desse tipo de aprendizado é construir um modelo  $(y = f(X))$  de uma função desconhecida,  $f$ , também chamada de função conceito, que permite prever valores de  $y$  para exemplos de  $X$  que ainda não foram vistos [24]. O aprendizado supervisionado ainda pode ser subdividido entre problemas de regressão, em que um valor numérico é previsto e problemas de classificação, nos quais o resultado da previsão é uma classe categórica [24].

Já no aprendizado não supervisionado, conforme apontam Janiesch et. al (2021), é fornecido ao algoritmo de aprendizado um conjunto de dados de treinamento ( $X$ ) e o mesmo é capaz de detectar padrões sem nenhum rótulo pré-existente [23]. Segundo Batista (2003), nesse tipo de aprendizado o principal objetivo é “construir um modelo que procura por regularidades nos exemplos, formando agrupamentos ou *clusters* de exemplos que compartilham características similares” [24].

No aprendizado por reforço, diferente das abordagens citadas anteriormente, não são fornecidos quaisquer exemplos de entrada ou saída. Janiesch et. al (2021) afirmam que neste tipo de aprendizado “descreve-se o estado atual do sistema, especifica-se um objetivo, é fornecido uma lista de ações permitidas, restrições ambientais e permite-se que o modelo de aprendizado de máquina experimente o processo de alcançar o objetivo por si só, usando o princípio de tentativa e erro”. Ou seja, durante o processo de aprendizado, são recebidas recompensas ou penalidades pelas ações realizadas, de forma que o objetivo é tomar decisões sequenciais para maximizar

a recompensa total. Esse tipo de aprendizado é amplamente aplicado em locais com ambiente dinâmico, como jogos de computador, robótica, entre outros [23].

Goodfellow et. al (2016) afirmam que inicialmente o campo do aprendizado de máquina conseguiu resolver problemas que eram intelectualmente difíceis para os seres humanos, porém simples para os computadores, como por exemplo, identificar regras matemáticas a partir de dados [25]. No entanto, o verdadeiro desafio deste campo é executar tarefas que são fáceis para os seres humanos. Geralmente são “problemas” que são resolvidos intuitivamente, como reconhecer palavras ou rostos em imagens. Para suprir esta demanda surgiu o campo do Aprendizado Profundo, ou *Deep Learning* [25].

Segundo Kelleher (2019), *Deep Learning* é uma subárea da inteligência artificial e do aprendizado de máquina focada na criação de modelos de redes neurais (*Neural Network*) que são capazes de aprender a partir de uma grande quantidade de dados. De acordo com Ekman (2021), a parte fundamental do *Deep Learning* é a *Deep Neural Network* (DNN), ou seja, as Redes Neurais Profundas, que foram inspiradas no neurônio biológico [26][27]. A Figura 1, a seguir, apresenta o relacionamento entre as áreas.

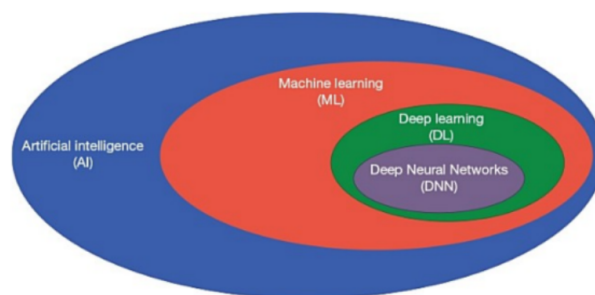


Figura 1. Áreas da Inteligência Artificial. Fonte: [26]

As redes neurais constituem a base dos algoritmos de aprendizado profundo. Como se pode observar na Figura 2 a seguir, as mesmas são compostas por três tipos de camadas: uma camada de entrada (*input layer*), uma ou mais camadas ocultas (*hidden layer*) e uma camada de saída (*output layer*) [28].

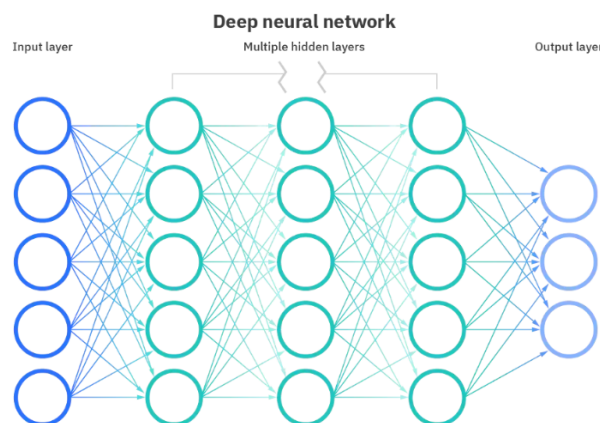


Figura 2. Estrutura da Rede Neural. Fonte: [29]

As redes neurais profundas são assim chamadas justamente pela sua complexidade em relação ao número de camadas. A “profundidade” dessas redes refere-se justamente a essa característica de possuir diversas camadas ocultas [29]. De acordo com Hardesty (2017), os avanços nos hardwares modernos possibilitaram a expansão das redes de apenas duas ou três camadas, nas décadas de 1980, para até 50 camadas nos dias atuais [30]. Desse modo, tal avanço tecnológico tem impulsionado a capacidade das redes neurais profundas em lidar com problemas em grande escala e cada vez mais complexos [30].

Atualmente, há três tipos popularmente conhecidos de arquiteturas de redes neurais profundas: Perceptrons Multicamadas (MLP), Redes Neurais Convolucionais (CNN) e Redes Neurais Recorrentes (RNN) [31].

A Perceptrons Multicamadas é o tipo mais básico de rede neural, composta por camadas totalmente conectadas entre si, conforme pode ser observado na Figura 2 e são muito utilizadas na resolução de problemas de classificação e regressão [31]. Já as Redes Neurais Convolucionais são utilizadas no processamento de dados estruturados em grades, como imagens. Essas redes extraem automaticamente características relevantes dos dados em cada camada e são amplamente utilizadas em tarefas de visão computacional como classificação de imagens, autenticação facial e segmentação de imagens [31].

As Redes Neurais Recorrentes foram projetadas para lidar com dados sequenciais, como texto ou séries temporais. Essas redes são formadas por neurônios que possuem memória e são capazes de armazenar informações anteriores, e suas conexões formam um gráfico de uma sequência temporal. As RNNs são comumente utilizadas em tarefas de Processamento de Linguagem Natural (NLP), permitindo a compreensão e geração de texto, tradução automática e análise de sentimento, entre diversas outras aplicações [31].

### C. Processamento de Linguagem Natural

O Processamento de Linguagem Natural, do inglês *Natural Language Processing* (NLP), também é uma subárea da Inteligência Artificial relacionada à representação e análise computacional da linguagem humana [32]. Khurana et al. (2023) afirmam que “a NLP é um tratado de Inteligência Artificial e Linguística, dedicado a fazer com que os computadores entendam as declarações ou palavras escritas em linguagens humanas. Surgiu para facilitar o trabalho do usuário e satisfazer o desejo de se comunicar com o computador em linguagem natural” [33].

Segundo Liddy (2001), a pesquisa em NLP começou no final dos anos 1940, sendo que a tradução automática foi a primeira aplicação da tecnologia. Entre os anos 60 e 70, houve um progresso significativo com o desenvolvimento de alguns sistemas nesta área de pesquisa. A ELIZA, que já foi citado anteriormente na subseção A, foi projetada para imitar uma conversa entre psicólogo e paciente, enquanto que SHRDLU, desenvolvido por Terry Winograd, era um programa que simulava um robô manipulando objetos em um ambiente virtual. Tais sistemas, apesar de suas limitações, demonstraram o potencial da máquina em conseguir compreender a linguagem humana, o que impulsionou diversos avanços e pesquisas na área nos anos seguintes [34].

Conforme aponta Liddy (2001), o crescimento acelerado da área de NLP nas últimas décadas foi impulsionado por três fatores principais: o aumento da disponibilidade de textos em formato digital, a evolução dos hardwares dos computadores (velocidade e capacidade de memória) e a Internet [34].

Conforme apontam Khurana et al. (2023), a NLP pode ser classificada em duas partes: Compreensão da Linguagem Natural ou *Natural Language Understanding* (NLU) e a Geração da Linguagem Natural ou *Natural Language Generation* (NLG) [33].

“O NLU permite que as máquinas entendam a linguagem natural e a analisem, extraindo conceitos, entidades, emoções, palavras-chave, etc” [33]. Este componente na NLP está diretamente associado à ciência da Linguística e seus diferentes segmentos, tais como Fonologia, Morfologia, Léxico, Sintaxe, Semântica etc. Já a NLG “é o processo de produção de frases, sentenças e parágrafos a partir de uma representação interna” [33].

Segundo Aranha (2007), uma etapa fundamental no processamento de linguagem natural é o pré-processamento do texto, que “consiste em um conjunto de transformações realizadas sobre alguma coleção de textos com o objetivo de fazer com que esses passem a ser estruturados em um representação atributo-valor” [35].

Aranha (2007) afirma que a etapa de pré-processamento é extremamente necessária para melhorar a qualidade dos dados disponíveis e organizá-los para serem submetidos aos algoritmos de *machine learning*. Desse modo, as transformações promovidas por tal etapa consistem em “identificar, compactar e tratar dados corrompidos, atributos irrelevantes e valores desconhecidos” [35]. Algumas dessas etapas serão explicadas detalhadamente a seguir.

A etapa de tokenização consiste no processo de segmentação de um texto por meio de caracteres como espaço, vírgula e ponto, formando grupos que são denominados *tokens*. Tanto os grupos de caracteres quanto os delimitadores tornam-se *tokens*, o único elemento que é descartado é o espaço em branco. Por exemplo na frase: “O jogador, que está de camisa verde, marcou o gol da vitória.”, após a tokenização ficaria [O] [jogador] [,] [que] [está] [de] [camisa] [verde] [,] [marcou] [o] [gol] [da] [vitória] [35].

A lematização, consiste na etapa de simplificação do texto e, segundo Aranha (2007), tem como objetivo converter para a forma primitiva as múltiplas formas de representação das palavras, por exemplo, as palavras “livros” e “livraria” seriam substituídas por “livro” [35].

Outra etapa importante é a eliminação de palavras comuns (*stop-words*), que são aquelas palavras que não agregam nenhum significado semântico. Normalmente são artigos, verbos auxiliares, preposições, etc. Assim, com a sua eliminação é possível obter-se uma representação reduzida do texto [35]. Por exemplo, considerando frase: “O jogador, que está de camisa verde, marcou o gol da vitória.”, após tal etapa ficaria “jogador camisa verde marcou gol vitória”.

De acordo com Liddy (2001), o Processamento de Linguagem Natural possui inúmeras aplicações em diferentes campos, como: recuperação e extração de informações, sistemas de *chatbot*, sumarização, tradução automática, etc [34].

#### D. Chatbots

Na Figura 3, é possível visualizar uma linha do tempo que representa o histórico de criação dos *chatbots*, desde o início até os dias atuais.

Como já dito anteriormente, o primeiro *chatbot* desenvolvido foi a ELIZA, em 1966, que agia como uma psicoterapeuta. Sua capacidade de comunicação era bem limitada, mas serviu de inspiração para criação de outros *chatbots* futuramente. Em 1972, foi desenvolvido o PARRY, que simulava um paciente com esquizofrenia. O *Jabberwacky* surgiu em 1988 e foi escrito na linguagem *CleverScript*, que, por ser baseada em planilhas, propiciou uma maior facilidade para o desenvolvimento. Em 1995, foi criada a ALICE (*Artificial Linguistic Internet Computer Entity*), que foi o primeiro *chatbot* inspirado em ELIZA e foi desenvolvida na linguagem AIML (*Artificial Intelligence Markup Language*). A ALICE conversava sobre qualquer tópico e ganhou o prêmio Loebner de melhor programa de computador semelhante ao humano [5].

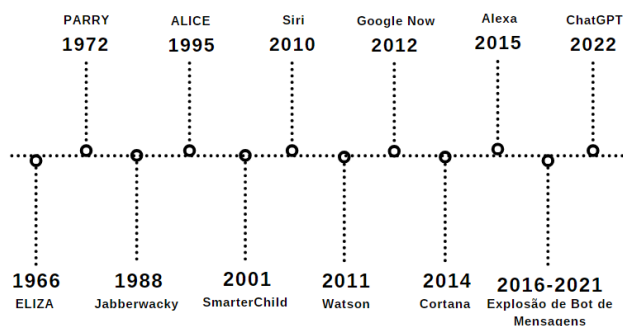


Figura 3. Linha do tempo da evolução dos bots de 1966 até 2022. Fonte: Autora

O *SmarterChild* surgiu em 2001 e estava disponível em mensageiros como o MSN. Foi o primeiro *chatbot* informacional que ajudava as pessoas em tarefas do dia a dia, sendo capaz de disponibilizar informações sobre horários de filmes, resultados esportivos, preços de ações, notícias e previsão do tempo. A partir de 2010 começaram a surgir os assistentes de voz pessoais inteligentes, como a *Siri* (*Apple*), *Watson* (*IBM*), *Google Assistant*, *Microsoft Cortana* e *Amazon Alexa*. A partir de 2016, com a evolução da Inteligência Artificial, as plataformas online começaram a desenvolver *chatbots* para sua marca ou serviço, como uma forma de atendimento a seus clientes de modo que atualmente os *chatbots* já se propagaram para todos os campos, estando presentes em diversos setores e para diferentes fins, tais como *Marketing*, Sistemas de Apoio, Cuidados de Saúde, Entretenimento, Educação, etc [5].

Segundo Adamopoulou e Moussiades (2020), existem duas abordagens no desenvolvimento de um *chatbot*, que são a correspondência de padrões e o aprendizado de máquina [5].

Os *chatbots* baseados em correspondência de padrões, também conhecidos como *chatbots* baseados em regras, utilizam algoritmos de correspondência de padrões. Esses algoritmos analisam e classificam as entradas dos usuários de acordo com um conjunto de regras pré-definidas. A partir disso, é possível

selecionar a resposta mais adequada dentro de um conjunto preestabelecido. De acordo com Adamopoulou e Moussiades (2020), há diversas limitações nessa abordagem, podendo-se destacar a impossibilidade desses sistemas gerarem novas respostas, pois os mesmos apenas operam dentro do conjunto limitado. Além disso, é necessário um grande banco de dados com regras para conseguir cobrir efetivamente todas as possibilidades de interações e, ainda assim, seria inviável tratar erros gramaticais e de sintaxe dos usuários. Em contrapartida a isso, Adamopoulou e Moussiades (2020) ressaltam que tais *chatbots* possuem um tempo de resposta rápido pois “não é realizado um exame sintático ou semântico mais profundo do texto de entrada”[5].

Já os *chatbots* que adotam a abordagem de aprendizado de máquina extraem o conteúdo da entrada do usuário empregando técnicas de processamento de linguagem natural (NLP), além disso não é necessária uma resposta predefinida para cada possível entrada, pois ele consegue efetuar a geração de texto [5]. O *chatbot* mais popular atualmente que utiliza essa abordagem é o ChatGPT, desenvolvido pela OpenAI e lançado em novembro de 2022 [36].

Além destas, vale destacar os *chatbots* que possuem uma abordagem híbrida, a exemplo do presente trabalho. Neste caso, é utilizado o processamento de linguagem natural para extrair as informações da entrada do usuários, no entanto estes não são capazes de gerar texto, então utiliza-se respostas predefinidas para cada tipo de *input* dos usuários.

### III. TRABALHOS CORRELATOS

A seguir são apresentados alguns trabalhos que se correlacionam com o presente projeto. No entanto, vale destacar que apesar dos mesmos possuírem objetivos e abordagens semelhantes, foram idealizados para atender a demandas específicas dos ambientes acadêmicos para os quais foram desenvolvidos.

#### A. Yuri

O *chatbot* Yuri foi desenvolvido por um estudante do Instituto Federal do Espírito Santo (IFES) - Campus Cachoeiro de Itapemirim e tem como objetivo principal a facilitação do acesso às informações relativas aos Processos Seletivos e Programas de Assistência Estudantil do campus. A interface do *chatbot* pode ser acessada através de uma página web, e, além da tela de interação com o assistente virtual, o sistema também conta com uma painel administrativo para que usuários administradores possam aperfeiçoar e configurar o funcionamento do bot [8].

A Figura 4 apresenta a tela inicial do Yuri. Ao clicar no botão “Fale Comigo”, abre-se uma pequena janela de conversa que permite a interação com o assistente. Vale destacar que essa tela é responsiva, ou seja, ela se adapta ao formato de tela dos aparelhos *mobile*, o que é um ponto positivo, pois torna-se mais acessível para que os estudantes possam utilizá-lo em seus dispositivos móveis.

Já a Figura 5 apresenta a tela de criação das intenções do usuário (*intents*), que consiste em uma das telas do Painel Administrativo do Yuri.

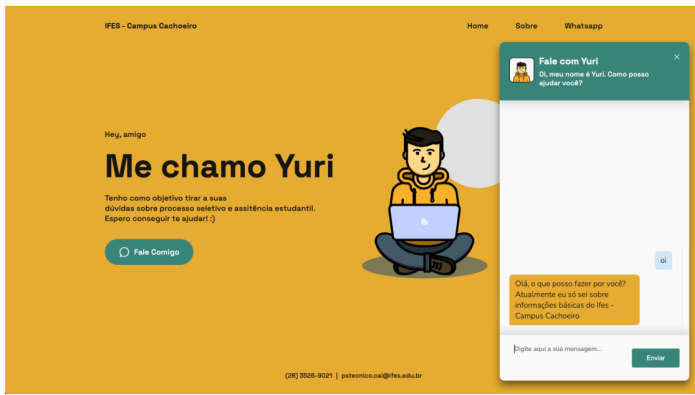


Figura 4. Tela de conversão do Yuri. Fonte: [8]

O painel administrativo do Yuri conta com as seguintes funcionalidades:

- Listagem das intenções de usuário cadastradas;
- Listagem das mensagens dos usuários que o Yuri não conseguiu responder;
- Realizar o treinamento do modelo do *chatbot*.

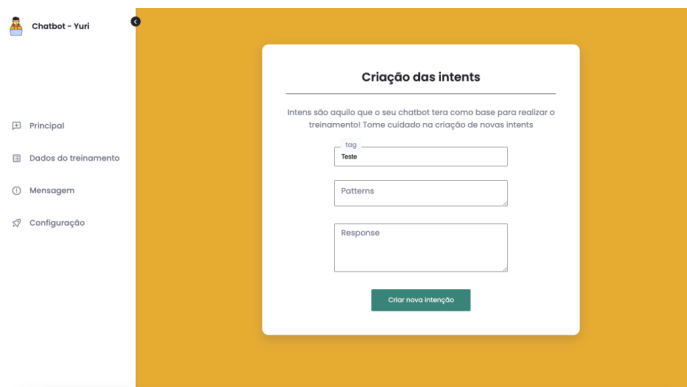


Figura 5. Tela do painel administrativo do Yuri. Fonte: [8]

Apesar da ótima proposta, o painel administrativo do Yuri apresenta alguns problemas, dentre eles pode-se destacar: a ausência de filtros e paginação nas telas de listagem das intenções de usuário e listagem de mensagens não respondidas. Tal fato dificultaria o controle e visualização caso houvesse muitos dados para serem exibidos. Além disso, a tela de cadastro de intenções de usuário exige que o administrador faça a inserção das informações no formato da estrutura de dados do tipo “lista” da linguagem de programação *Python*, o que não é nada intuitivo e seria um elemento dificultador caso o administrador fosse um usuário leigo na área de programação.

Já em relação à abrangência de conhecimento, como já dito anteriormente, o Yuri possui capacidade de responder a perguntas relacionadas aos Processos Seletivos e Programas de Assistência Estudantil do IFES - Campus Cachoeiro de Itapemirim. Sendo assim, seria mais voltado para a população externa que tem interesse em ingressar no instituto do que para

o público interno, diferentemente do presente projeto cujo foco é a comunidade discente do IFBA - Campus Salvador.

## B. RegBot

O *chatbot* RegBot foi desenvolvido por uma estudante da Universidade Federal de Campina Grande (UFCG) e tem como objetivo principal possibilitar ao discente da instituição o acesso às informações sobre o Regulamento de Ensino de Graduação de forma mais fácil e ágil, além de disponibilizar os dados em uma linguagem mais acessível. A interface do *bot* pode ser acessada através de uma página web [2].

A Figura 6 apresenta as telas de conversão do RegBot, que possui um *design* bem simples sem muitos recursos. Como pode-se observar na imagem, além da conversa por texto, também é possível mandar mensagens de áudio, o que acrescenta mais praticidade ao assistente, visto que atualmente as pessoas estão habituadas a se comunicarem por meio de mensagens de voz. Como já dito, o sistema possui um área de atuação bem delimitada, dado que seu foco é apenas no regulamento acadêmico da UFCG.

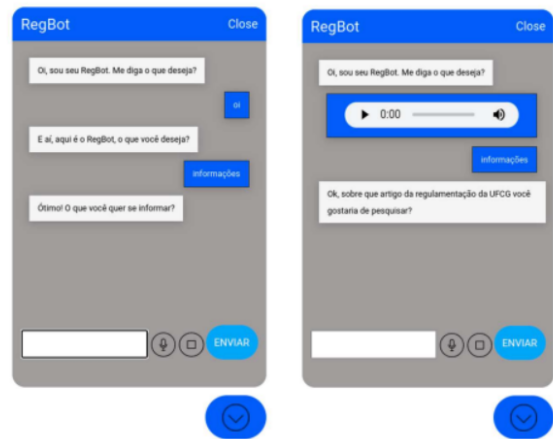


Figura 6. Tela de conversação do RegBot. Fonte: [2]

## C. Ifes.talk

O *chatbot* denominado “Ifes.talk” foi desenvolvido por um estudante do Instituto Federal do Espírito Santo (IFES) - Campus Serra e tem o propósito de oferecer suporte aos discentes do instituto, auxiliando-os em suas dúvidas relacionadas às rotinas acadêmicas. O projeto visa simplificar a busca por informações no ROD (Registro de Orientação Didática), proporcionando respostas claras e diretas, por meio de uma linguagem mais acessível [6].

Diferentemente das soluções explanadas anteriormente, o Ifes.talk é acessado através dos aplicativos de mensagens *Facebook Messenger* e *Telegram*, pois segundo o seu desenvolvedor são aplicativos amplamente utilizados pelos discentes do campus. Além disso, o *bot* também conta com um painel administrativo web, cujo objetivo é facilitar a gestão do *chatbot* tanto por membros do projeto quanto por servidores do IFES, mesmo sem conhecimento avançado em tecnologia, graças à sua interface intuitiva. No painel é possível editar as respostas do *chatbot*, acessar estatísticas de uso, visualizar *feedbacks*

gerais dos usuários sobre a ferramenta, etc. As telas do sistema podem ser observadas a seguir [6].

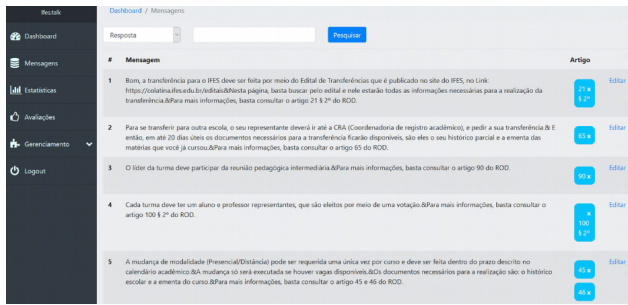


Figura 7. Painel Administrativo Ifes.talk. Fonte: [6]

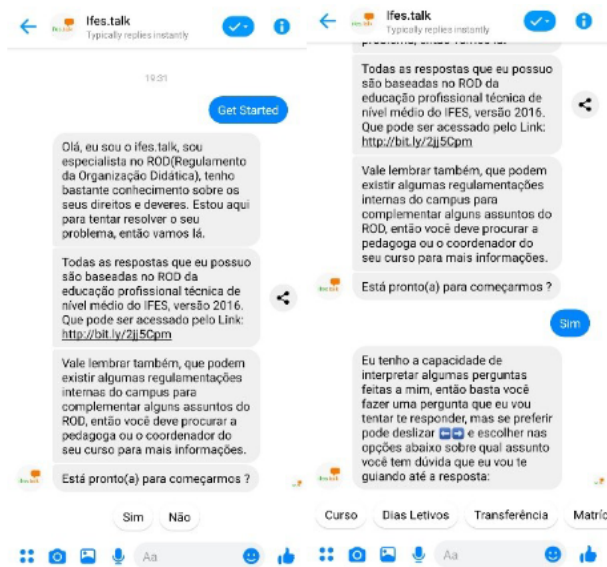


Figura 8. Conversa do Ifes.talk no Facebook Messenger. Fonte: [6]

#### D. Análise Comparativa

Todas as aplicações explanadas anteriormente possuem pontos em comum e diferenças de funcionamento em relação à solução desenvolvida neste trabalho. A seguir encontra-se a Figura 9 que apresenta uma tabela comparativa de tais trabalhos e o IFBot, fazendo uma síntese das principais semelhanças e diferenças.

Conforme ilustrado na Figura 9, o IFBot se sobressai em diversas funcionalidades, destacando-se em comparação com outros sistemas. Uma das principais vantagens do IFBot é a sua interface de conversação adaptável a dispositivos móveis, permitindo uma experiência mais ampla e acessível aos usuários, sem necessidade de utilizar qualquer aplicativo para ter acesso ao chat, diferentemente do Ifes.talk. Além disso, oferece um painel administrativo robusto com a capacidade de editar e criar novas respostas para o chatbot, efetuar o treinamento do modelo, acessar as perguntas não respondidas e as respostas do bot que tiveram feedback negativo, sendo que todas estas funcionalidades contam também com filtros personalizados, que é uma vantagem significativa para o gerenciamento prático dos dados do sistema à medida que estes escalam.

Funcionalidades	IFBot	Yuri	RegBot	Ifes.talk
Interface de conversação adaptável a dispositivos móveis	✓	✓	✓	✗
Necessita de um aplicativo de mensagens para utilização do chat	✗	✗	✗	✓
Suporte a mensagens de áudio	✗	✗	✓	✗
Painel Administrativo	✓	✓	✗	✓
Gerenciamento dos dados com filtros personalizados no painel administrativo	✓	✗	✗	✓
Possibilidade de feedback do usuário em cada respostas do chatbot	✓	✗	✗	✗
Possibilidade de feedback do usuário sobre a utilização do sistema	✗	✗	✗	✓

Figura 9. Quadro comparativo entre IFBot, Yuri, RegBot e Ifes.talk. Fonte: Autora

## IV. SOLUÇÃO DESENVOLVIDA

A presente seção tem como objetivo detalhar a solução desenvolvida, apresentando o processo de criação da base de dados do chatbot, as tecnologias utilizadas em sua construção, a arquitetura, os requisitos e respectivos diagramas.

O propósito desse trabalho foi construir um chatbot que visa aprimorar a experiência dos estudantes do IFBA - Campus Salvador ao buscar dados do cotidiano acadêmico, tais como informações sobre os cursos, matrícula, trancamento, ementas, aproveitamento de disciplinas, localizações de salas de aula, etc. O intuito principal é viabilizar uma ferramenta que permita uma maior agilidade e conveniência para obter informações e auxiliar na resolução de dúvidas comuns. A seguir, a Figura 10 apresenta uma visão geral dos componentes do sistema.

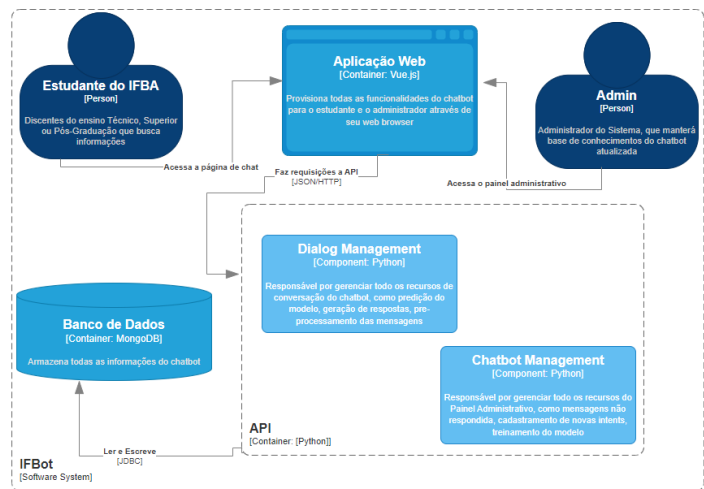


Figura 10. Diagrama com os componentes principais do IFBot. Fonte: Autora

Conforme pode ser observado na Figura 10, o software possui três componentes principais:

**Aplicação Web:** Este é o componente de interface acessível aos usuários através de um navegador web. A aplicação web é responsável por provisionar todas as funcionalidades do chatbot, tais como a tela de chat para os estudantes, como

também o painel administrativo para os usuários que irão gerenciar o IFBot.

**API:** A API (*Application Programming Interface*, ou Interface de Programação de Aplicações) representa o *backend* da aplicação e é subdividida em diferentes módulos que gerenciam duas áreas cruciais: o *Dialog Management* (Gerenciamento de Diálogo), que é responsável pelas funcionalidades do *chat* e o *Chatbot Management* (Gerenciamento do *Chatbot*), que provisiona as funcionalidades do Painel Administrativo.

**Banco de Dados:** Este componente é responsável por armazenar todas as informações relacionadas ao sistema de *chatbot*, onde são guardados dados essenciais para sua execução.

O fluxo de funcionamento do IFBot se dá através da interação entre os componentes em que a Aplicação Web se comunica com a API, que, por sua vez, se conecta ao Banco de Dados para obter ou armazenar informações conforme necessário.

A seguir, serão detalhados cada um desses componentes, bem como serão abordados o processo de criação da base de dados e o treinamento do modelo.

#### A. Base de Conhecimentos

A primeira etapa para o desenvolvimento de um *chatbot* é identificar as possíveis questões que os usuários poderão ter e assim, a partir destas elaborar as respostas que serão fornecidas a eles. Para isso é necessária a criação das intenções de usuários, também denominadas de *intents*. As *intents* são conceituadas por Singh (2023) como a motivação de um usuário do *chatbot*, ou seja, é o propósito por trás da mensagem que o assistente virtual recebe [37].

No presente trabalho, as principais fontes de dados para a criação das *intents* do *chatbot* foram os documentos institucionais do IFBA, além de outras informações disponíveis no portal online do Campus Salvador, tais como:

- Normas Acadêmicas do Ensino Superior;
- Organização Didática do Ensino Técnico;
- Dados de processos seletivos;
- Informações sobre assistência estudantil;
- Informações de contatos do campus;
- Localizações de salas e departamentos;
- Calendário acadêmico;
- Horários de aulas;
- Informações sobre os cursos (fluxogramas, projetos pedagógicos, estrutura curricular);

Além disso, visando obter uma melhor compreensão sobre as dúvidas recorrentes dos estudantes dos Institutos Federais, utilizou-se as perguntas contidas nas páginas de “Dúvidas Frequentes” tanto do portal do Campus Salvador, quanto dos sites de outros Institutos Federais Brasileiros.

Inicialmente, as *intents* foram estruturadas no formato JSON (*JavaScript Object Notation*), que é uma estrutura de

dados originada da linguagem de programação *JavaScript* e é muito utilizada para armazenamento e transmissão de dados [38]. A estrutura de cada *intent* possui os atributos *category*, *tag*, *patterns*, *responses* e *context*, conforme pode ser observado na Figura 11 a seguir.

```
{
  "intents": [
    {
      "category": "personalidade_bot",
      "tag": "saudacao",
      "patterns": [
        "Oi",
        "Ei",
        "Tem alguém aí?",
        "Olá",
        "Opa",
        "Hello",
        "Bom dia",
        "Boa tarde",
        "Boa noite",
        "Hey",
        "Como vai?"
      ],
      "responses": [
        "Olá, como posso te ajudar? Atualmente eu sei sobre:"
        "\n★ Normas Acadêmicas do Ensino Técnico e Superior"
        "\n★ Informações sobre o processo seletivo do Campus Salvador"
        "\n★ Informações sobre os Cursos do Campus Salvador"
        "\n\nTodas as respostas que possuo são baseadas nas Normas Acadêmicas "
        "do Ensino Superior, na Organização Didática do Ensino Técnico e outras "
        "informações contidas no site do IFBA - Campus Salvador"
        "Eu tenho a capacidade de interpretar suas perguntas, mas se preferir "
        "na barra lateral há alguns sugestões de assuntos, você pode escolher e eu "
        "vou te responder com base em meus conhecimentos. Vamos lá? 🍷",
        ""
      ],
      "context": [""]
    }
  ]
}
```

Figura 11. Exemplo de intenção de usuário do IFBot. Fonte: Autora

O atributo *category* tem a finalidade de agrupar cada intenção em categorias pré-definidas de acordo com o tema da intenção (saudações, matrícula, horários, informações sobre cursos, etc) para que os administradores consigam filtrar adequadamente as *intents*, de forma a facilitar o gerenciamento dos dados à medida que estes vão escalando. O atributo *tag* consiste no identificador escrito da intenção e é nela que está descrito o propósito do usuário ao se comunicar com o *chatbot*. Já os *patterns* são constituídos por uma lista em que estão enumeradas as várias frases que os usuários podem utilizar para se expressar. O atributo *responses* consiste nas possíveis respostas definidas para o *bot*, e o *context* é onde podem ser listadas *tags* de outras *intents* que estão relacionadas ao contexto da atual.

Após o levantamento de todas as intenções de usuário, o documento JSON contendo-as foi inserido no banco de dados *MongoDB*, que será explanado na seção a seguir.

#### B. Armazenamento de Dados

Para o armazenamento dos dados foi utilizado o banco não relacional *MongoDB*, um dos mais conhecidos na sua categoria por sua rapidez. Segundo Zhao et al. (2013), o *MongoDB* é um banco de dados de código aberto NoSQL do tipo orientado a documentos. Foi projetado para desempenho excepcionalmente alto e desenvolvido em C++ [39]. Atualmente o *MongoDB* é utilizado por mais de 47.800 empresas, dentre elas pode-se destacar: Toyota, Bosch, TIM, Forbes, Electrolux [40].

Os dados no *MongoDB* são armazenados em formato BJSON, que é um análogo à estrutura de dados JSON. Conforme apontam Zhao et al. (2013), o *MongoDB* tem suporte às seguintes funcionalidades:

- Consultas *Ad hoc* (Por se um banco orientado a documentos não há possibilidade de executar transações nem fazer *joins*, desse modo, a escrita das consultas é bem simples);
- Replicação (Suporta a arquitetura de replicação *master-slave*);
- Armazenamento de Arquivos.

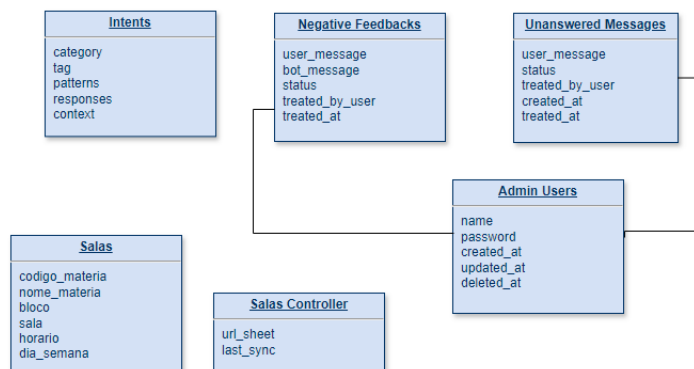


Figura 12. Diagrama que representa a estrutura do banco de dados. Fonte: Autora

O diagrama ilustrado na Figura 12 oferece uma visão detalhada sobre a estrutura das *collections* que compõem o banco de dados do sistema IFBot. *Collections* são agrupamentos de documentos do *MongoDB*, equivalente às tabelas no banco de dados relacional [41]. A seguir, há uma descrição de cada uma e suas funções dentro desse sistema:

- *Intents*: Armazena as intenções de usuário, conforme já abordado na subseção IV-A exposta anteriormente;
- *Negative Feedbacks*: Responsável por registrar as mensagens enviadas pelo IFBot que receberam *feedback* negativo dos usuários. O armazenamento dessas interações é essencial para a análise e o aprimoramento contínuo do sistema, permitindo identificar e corrigir falhas ou imprecisões nas respostas fornecidas;
- *Unanswered Messages*: Responsável por registrar as mensagens não respondidas, ou seja, perguntas dos usuários para os quais o IFBot não conseguiu fornecer uma resposta. O monitoramento dessas lacunas na capacidade de resposta do *bot* é fundamental para a expansão de seu conhecimento e habilidades;
- *Salas Controller* e *Salas*: São responsáveis por armazenar as informações relacionadas às localizações de salas de aula. A *collection* *Salas Controller* armazena o *link* para a planilha de alocação de salas, que é disponibilizada pela Diretoria de Ensino (DE) através do *Google Sheets*. Já a *collection* *Salas* é destinada ao armazenamento das informações extraídas dessas planilha.

### C. Classificador de Intents

Para que o *chatbot* forneça uma resposta para uma pergunta, é necessária a criação da funcionalidade que efetua

o mapeamento de uma a entrada do usuário para a *tag* de uma *intent*. Como um dos objetivos do presente trabalho é ter um bom desempenho nas respostas, não será utilizada a abordagem de efetuar uma correspondência exata entre a entrada do usuário e as frases cadastradas no atributo *patterns* de cada intenção. Ao invés disso serão utilizadas técnicas de Processamento de Linguagem Natural e *Deep Learning*. Dessa forma, as frases cadastradas anteriormente nos atributos *patterns* das *intents* serão utilizadas para efetuar o treinamento de um Modelo de Classificação de Intenções.

Para a criação do classificador foi utilizada a linguagem de programação *Python*, juntamente com a bibliotecas *spaCy* (para o pré-processamento das informações), *Tensorflow* e a *API Keras* para a construção da rede neural e treinamento do *bot*.

O *spaCy* é uma biblioteca de software comercial de código aberto escrita nas linguagens de programação *Python* e *Cython* e desenvolvida por Matthew Honnibal. É utilizada para processamento avançado de linguagem natural (PLN). A biblioteca foi desenvolvida com base em pesquisas mais recentes e foi projetada para ser empregada em produtos reais, diferentemente da *NLTK* (*Natural Language Toolkit*), outra biblioteca do *Python* para PLN, que foi escrito principalmente como um recurso educacional. O *spaCy* tem suporte para mais de 73 idiomas (dentre estes o português) e possui modelos de rede neural para marcação, análise, reconhecimento de entidade nomeada, classificação de texto, além de muitos outros recursos [42][43].

A *pipeline* de pré-processamento dos dados, que é a etapa anterior ao treinamento e as previsões, obedeceu o fluxo mostrado na Figura 13, em que cada etapa será explanada a seguir.

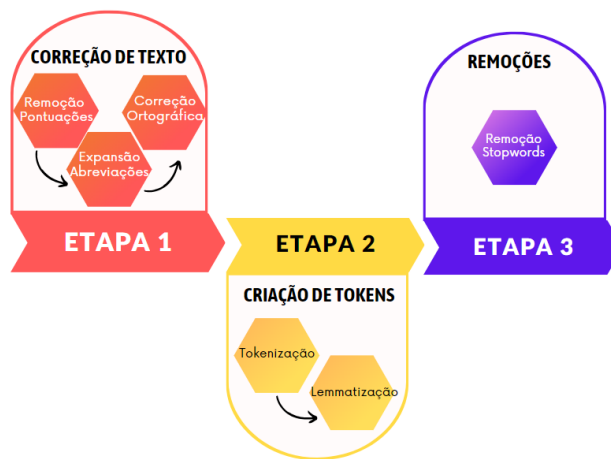


Figura 13. Etapas de pré-processamento do texto. Fonte: Autora

A Etapa 1 consiste na uniformização do texto. Dessa forma, efetuou-se a remoção de algumas pontuações, remoção de espaços em branco duplicados, expansão de palavras abreviadas (que é uma prática muito comum entre os jovens ao enviar mensagens virtualmente) e por fim é feita uma correção de erros ortográficos utilizando-se a biblioteca *SpellChecker* do *Python*, que possui suporte para a língua portuguesa.

A Etapa 2 consiste respectivamente na Tokenização (di-

visão do texto em palavras) e Lemmatização (remoção das flexões), que são realizadas utilizando-se a biblioteca *spaCy*, cujo tokenizador possui 100% de acurácia e o lematizador 97% [44].

Por fim, a Etapa 3 efetua a remoção das *stopwords*, que são palavras que podem ser consideradas irrelevantes, como artigos, conjunções, preposições, verbos auxiliares, etc. A biblioteca *spaCy* também contempla uma lista de *stopwords* da língua portuguesa. A seguir, na Figura 14, encontra-se um exemplo de um diagrama de estados que demonstra o fluxo de texto ao longo da pipeline de pré-processamento.

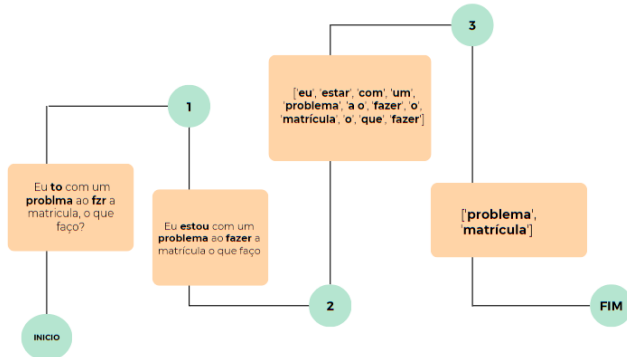


Figura 14. Diagrama com estados do texto ao longo da pipeline de pré-processamento. Fonte: Autora

Após o pré-processamento dos dados, é possível criar a rede neural e treinar o modelo. Para isso, utilizou-se o *TensorFlow* e a *API Keras*. O *TensorFlow* é um poderoso software de código aberto desenvolvido pelo time do *Google Brain* para criar modelos de *machine learning* e é compatível com diversas linguagens de programação como *Python*, *C++*, *Java*, etc. Já o *Keras* é uma API de código aberto escrita em *Python* e serve para criar e treinar modelos de *Deep Learning* de forma simplificada. O *Keras* pode ser integrado com múltiplos *deep learning engines*, dentre elas o *TensorFlow*. O código de treino encontra-se a seguir [45].

```

1 train_X , train_y = self.get_train_data()
2
3 self.model.add(Dense(128, activation='relu',
4   input_shape=(len(train_X[0])))
5 self.model.add(Dropout(0.5))
6 self.model.add(Dense(185, activation='relu'))
7 self.model.add(Dropout(0.5))
8 self.model.add(Dense(len(train_y[0]), activation='
9   softmax'))
10
11 # Monta pipeline de treino
12 self.model.compile(optimizer=self.optimizer, loss=
13   loss, metrics=['accuracy'])
14 weights = self.model.fit(
15   x=np.array(train_X),
16   y=np.array(train_y),
17   epochs=epochs,
18   batch_size=batch_size,
19   verbose=verbose
20 )
21
22 self.model.save(self.root_path / self.model_name,
23   weights)
  
```

Listing 1. Código treinamento Classificador de Intenções

Como pode-se observar no código exposto anteriormente, a rede neural foi construída utilizando-se três camadas (*Dense*), sendo uma de entrada, uma intermediária ou oculta e uma de saída. Optou-se por utilizar-se apenas uma camada oculta pois segundo Bishop (1995) não é recomendado a utilização de mais de duas camadas ocultas, sendo que para a maioria dos problemas de classificação apenas uma já é suficiente [46] [47] [48].

Vale destacar que como não foi utilizada uma grande quantidade de dados para efetuar o treinamento do modelo, utilizou-se o *Dropout* entre cada camada. *Dropout* “é uma técnica de regularização que consiste em desativar aleatoriamente um conjunto de neurônios durante o treinamento, forçando o modelo a aprender de maneira mais robusta e melhorando sua capacidade de generalização” [49]. Desse modo, é possível reduzir o *overfitting*, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento e assim não performa bem para dados que nunca foram vistos [47] [48].

#### D. Backend

Os componentes que gerenciam o *chatbot* foram desenvolvidos utilizando-se a linguagem de programação *Python* e a API foi construída utilizando-se o *FastAPI*.

O *FastAPI* é um *framework* de alto desempenho para a criação de APIs em *Python*, que se destaca por sua rapidez e por facilitar um desenvolvimento ágil. É caracterizado por ser intuitivo, fácil de aprender e vem com documentação interativa, tornando-o ideal para projetos que buscam eficiência e robustez. Atualmente, o *FastAPI* é usado por empresas como *Microsoft*, *Uber*, *Netflix*, *Sallve*, etc [50].

O *backend* foi elaborado para atender aos seguintes requisitos não funcionais:

Tabela I. REQUISITOS NÃO FUNCIONAIS *backend*

Código	Nome	Especificação
RNF001	Segurança	O <i>backend</i> deve implantar políticas de segurança para proteger os dados administrativos do <i>chatbot</i> e as credenciais dos usuários.
RNF002	Disponibilidade	O <i>backend</i> deve estar disponível na maior parte do tempo.
RNF003	Manutenibilidade	O <i>backend</i> deve possuir uma arquitetura bem definida para facilitar a manutenção contínua do sistema.
RNF004	Escalabilidade	A arquitetura do <i>backend</i> deve ser construída para suportar o crescimento do sistema de forma eficiente.

A arquitetura da API foi implementada visando os requisitos de manutenibilidade e escalabilidade, desse modo, a mesma possui duas categorias distintas de rotas, que atendem a diferentes partes do sistema. As rotas de *chat* estão relacionadas às funcionalidades de conversação do IFBot, as quais dispensam autenticação. Já as rotas *admin* são destinadas à interface administrativa. Por estas rotas estarem atreladas ao gerenciamento dos dados do sistema, seu acesso exige autenticação, garantindo assim a segurança e a integridade dessas informações. As imagens das rotas podem ser visualizadas nas Figuras 15 e 16 a seguir.

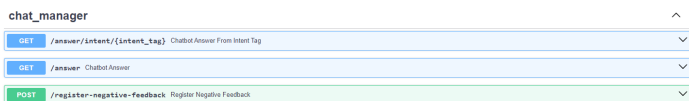


Figura 15. Rotas de *chat* da API do IFBot. Fonte: Autora

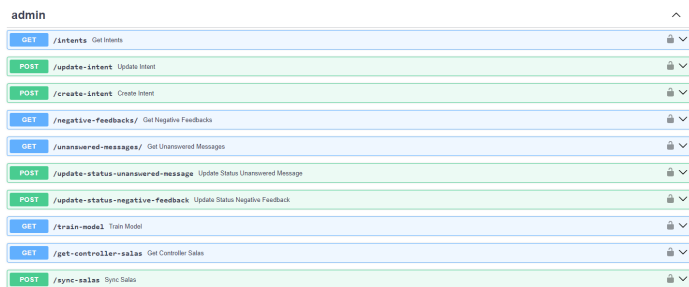


Figura 16. Rotas *admin* da API do IFBot. Fonte: Autora

Visando o requisito de disponibilidade, tanto o *backend* do sistema quanto o banco de dados *MongoDB* foram hospedados por meio do provedor de *cloud computing Amazon Web Service (AWS)*. Os detalhes da implantação serão abordados na subseção IV-F.

A segurança do sistema é baseada em *JWT (JSON Web Token)*, que consiste em um padrão aberto para criação de *tokens* de acesso que podem ser utilizados para autenticação e compartilhamento de informações de forma segura entre diferentes partes. Esse formato é baseado em *JSON* e é comumente utilizado para autenticação em sistemas *web* e *APIs* [51]. Nas rotas do *FastAPI*, o *token* é concedido após o login ser efetuado e precisa ser fornecido a cada requisição para assegurar a permissão de acesso aos dados [50].

Com o objetivo de mitigar possíveis vulnerabilidades decorrentes de exposições potenciais que possam vir a comprometer os dados dos usuários, as senhas dos usuários administrativos são criptografadas antes de serem armazenadas. Para isso é utilizado o método *Bcrypt*, através da biblioteca *Bcrypt Python*, que utiliza o algoritmo de *hash bcrypt*, considerado extremamente seguro. Reconhecido por sua robustez, o método *Bcrypt* adiciona um fator de custo à operação de *hash*, tornando extremamente difícil e demorado para que possíveis invasores acessem as senhas [52].

Os dois componentes que compõem o *backend* do sistema são: o *Chatbot Management*, responsável por gerenciar as funcionalidades das rotas administrativas e o *Dialog Management*, que é responsável por gerenciar as funcionalidades de *chat*. A arquitetura deste último com seus subcomponente pode ser visualizada na Figura 17 a seguir.

Como pode ser visualizado na Figura 17, a arquitetura de gerenciamento de diálogo do *chatbot* é formada por três componentes. O Componente de Análise da Mensagem de Usuário (*User Message Analysis Component*) é responsável por efetuar a análise da mensagem do usuário e identificar a intenção por trás dessa mensagem. É neste componente que ocorre o pré-processamento da mensagem de texto do usuário, que segue a *pipeline* exposta anteriormente na Figura 13. Após

isso, é executada a classificação da intenção de usuário através do Classificador de *Intents* no qual é retornado o resultado predito (*Intent Prediction*), ou seja, as intenções (*intents*) que foram inferidas com base na mensagem do usuário.

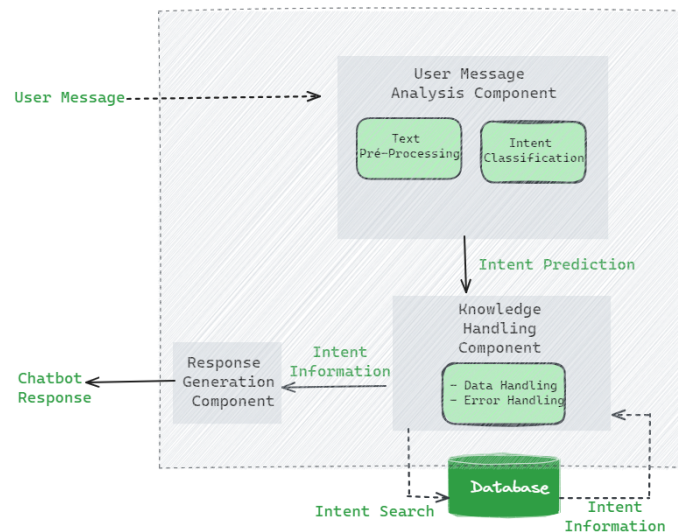


Figura 17. Subcomponentes do *Dialog Management*. Fonte: Autora

O Componente de Manipulação de Conhecimento (*Knowledge Handling Component*) é responsável por fazer todo o gerenciamento da base de conhecimento do *chatbot*. Sua função é buscar no banco de dados a *intent* correspondente com o resultado da predição de mais alta probabilidade feita pelo *User Message Analysis Component*. Além disso, é neste componente que ocorre o tratamento de erros, para os casos em que a identificação da intenção do usuário falha (gerando, assim, uma resposta padrão de não compreensão) ou quando as predições são efetuadas com uma probabilidade abaixo do valor pré-definido de 50% de confiança, gerando uma mensagem padrão que reflete a incerteza na resposta. Neste último caso as intenções que foram inferidas são apresentadas para os usuários como possibilidades de respostas. O diagrama de atividades exposto na Figura 18 demonstra o processo de execução do componente.

Já o Componente de Geração de Resposta (*Response Generation Component*) é responsável pela tarefa de criar a resposta que será apresentada ao usuário, utilizando os dados da intenção de usuário que foi recuperada do banco de dados anteriormente.

No que diz respeito às funcionalidades do painel administrativo, as mesmas são gerenciadas pelo componente *ChatBot Management*, que serão explanadas com mais detalhes na subseção IV-G. Vale ressaltar que esse componente também é responsável por extrair e tratar os dados da planilha de alocação de salas que é disponibilizada pela direção do Campus Salvador a cada semestre letivo. Para isso é utilizado a biblioteca do *Python gspread*, que é uma API do *Google Sheets* e através dela é possível acessar o conteúdo da planilha através da respectiva URL. Após a extração e organização, as informações são salvas no banco de dados para serem utilizadas como fonte de dados para que o IFBot responda perguntas associadas às localizações das salas.

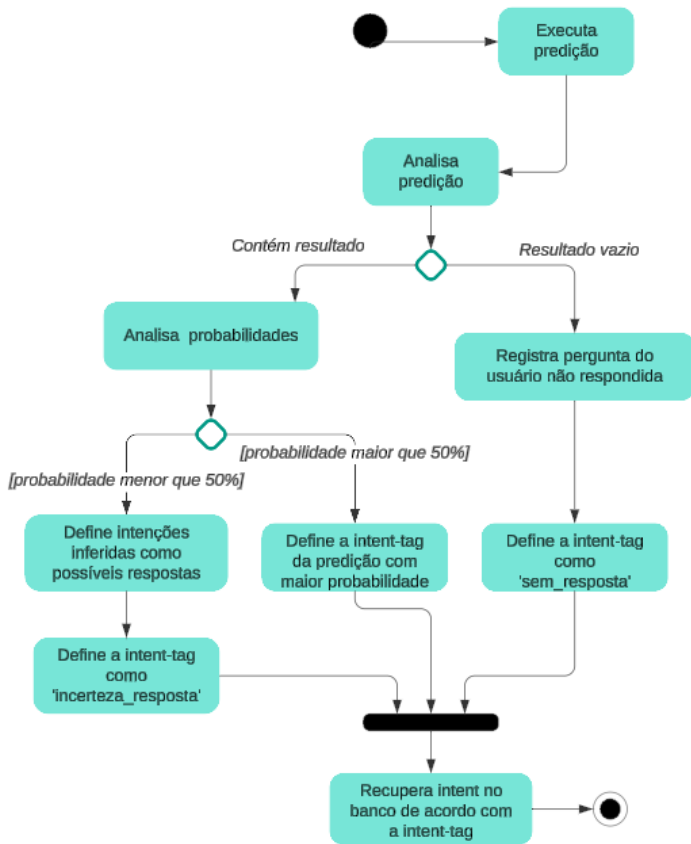


Figura 18. Diagrama de Atividades do *Knowledge Handling Component*. Fonte: Autora

### E. Aplicação Web

Para a construção da interface web foi utilizado o *Vue.js*, que é um *framework JavaScript* de código aberto para a construção de interfaces de usuário [53]. O *Vue.js* é um dos *frameworks JavaScript* mais usados em produção atualmente, com mais de 1,5 milhão de usuários em todo o mundo, incluindo grandes empresas como *NASA*, *Apple*, *Google*, *Microsoft*, *GitLab*, *Zoom*, entre muitas outras [53].

A estilização da aplicação web do IFBot foi feita através da utilização do *Vue Component Framework* denominado *Vuetify*. Ele é um projeto *Open Source* de *design* de interface baseado no *Material Design* da *Google* que oferece uma vasta gama de componentes prontos para uso e altamente personalizáveis [54].

A Figura 19 demonstra a forma que os pacotes da aplicação web estão organizados. Os componentes representam quaisquer elementos de interface do usuário, desde botões, formulários, barras de navegação. Eles promovem a reutilização de código e facilitam a manutenção da aplicação [55].

As *views* são componentes do *Vue.js* que representam as diferentes páginas ou seções da sua aplicação e são responsáveis por renderizar o conteúdo da página e interagir com o estado da aplicação. Cada *view* geralmente corresponde a uma rota específica definida pelo *router* [55].

O *router* é responsável pelo gerenciamento de rotas na

aplicação *Vue.js*, sendo que é através dele que é feito o mapeamento de cada rota para um componente específico que será renderizado quando a URL correspondente for acessada. Já o *store* é um gerenciador de estado e permite que o mesmo seja compartilhado entre as *views* e componentes [55].

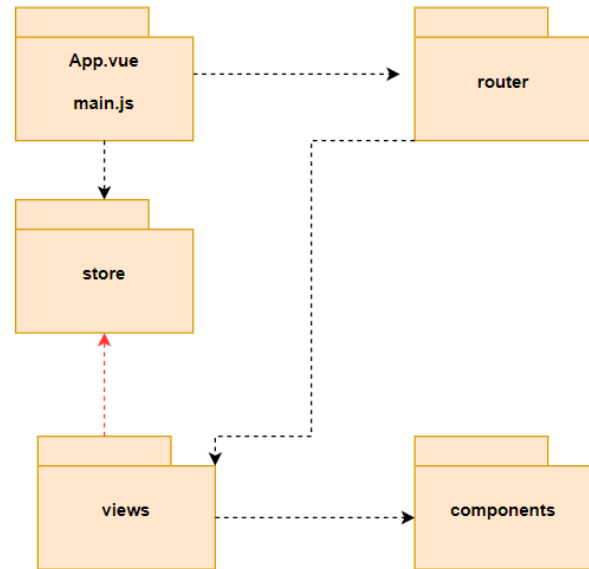


Figura 19. Diagrama de Pacotes da Aplicação Web. Fonte: Autora

### F. Implantação

De acordo com Sommerville (2016), a etapa final do processo de desenvolvimento de software consiste na entrega e implantação do produto [56].

O processo de *deploy* foi feito utilizando a *Amazon Web Services (AWS)*, que é um provedor de *Cloud Computing* (Computação em Nuvem) mais utilizada do mundo, possuindo milhões de clientes [57]. O serviço escolhido de hospedagem da aplicação foi o *Amazon Elastic Compute Cloud (Amazon EC2)* que possui mais de 750 instâncias e opções de processadores, armazenamentos, redes, sistemas operacionais, além de fornecer uma infraestrutura confiável e escalável sob demanda com 99,99% de disponibilidade [58].

### G. Funcionalidades

A aplicação foi feita visando atender aos seguintes requisitos funcionais, expostos na Tabela II.

Os requisitos estabelecidos para o sistema, possibilitam que o *frontend* tenha uma estrutura bem definida e atenda às necessidades dos usuários e administradores.

A Figura 20 apresenta a tela inicial do *chatbot*, que serve como o ponto de partida para os usuários interagirem com o sistema. Nesta tela, nesta tela há uma descrição geral sobre o IFBot e o botão “Fale Comigo”, que redireciona os usuários para a tela de *chat*.

Tabela II. REQUISITOS FUNCIONAIS DA INTERFACE WEB

Código	Nome	Especificação
RF001	Interagir com o <i>chatbot</i>	O sistema deve disponibilizar uma interface de <i>chat</i> que permita ao usuário fazer perguntas e receber as respectivas respostas.
RF002	Feedback da resposta do <i>chatbot</i>	A interface de <i>chat</i> deve permitir que o usuário forneça um <i>feedback</i> negativo caso a resposta do <i>chatbot</i> não tenha sido satisfatória.
RF003	Registro de perguntas não respondidas	O sistema deve salvar a pergunta do usuário que o <i>chatbot</i> não conseguiu fornecer uma resposta.
RF004	Área Administrativa	O sistema deve possuir uma área administrativa, com acesso restrito por meio de usuário e senha.
RF005	Cadastro e Atualização das Intenções de Usuário	O sistema deve fornecer ao administrador a possibilidade de cadastrar e atualizar as intenções de usuário para aumentar e manter atualizada a base de conhecimento do <i>chatbot</i> .
RF006	Visualizar Base de Dados	O sistema deve fornecer ao administrador a possibilidade de visualizar todas as intenções de usuário.
RF007	Filtragem da Base de Dados	O sistema deve fornecer ao administrador filtros personalizados para fazer buscas na base de dados.
RF008	Visualizar <i>feedback</i> das respostas	O sistema deve fornecer ao administrador a possibilidade de visualizar as perguntas e suas respectivas respostas em que o usuário forneceu um <i>feedback</i> negativo.
RF009	Visualizar perguntas não respondidas	O sistema deve fornecer ao administrador a possibilidade de visualizar as perguntas que não foram respondidas pelo <i>chatbot</i> .
RF010	Treinar o <i>Chatbot</i>	O sistema deve fornecer ao administrador a possibilidade de treinamento do <i>chatbot</i> .
RF011	Integração com Planilha de alocação de salas	O sistema deve permitir a importação dos dados de localizações e horários de disciplinas a partir da URL da planilha de alocação de salas disponibilizada pela DIREC.

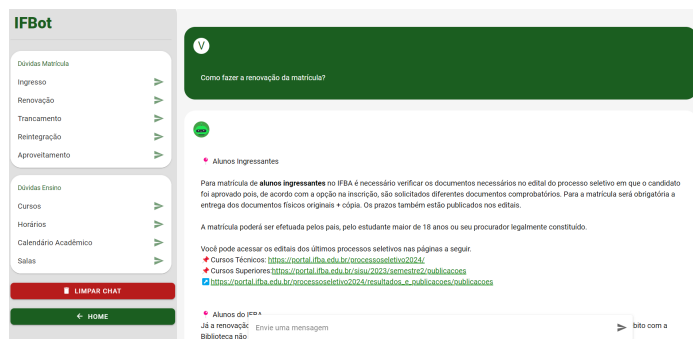


Figura 21. Tela de *chat* do IFBot versão *web*. Fonte: Autora

Outra função relevante que também está presente na tela de *chat* é a possibilidade do usuário avaliar a resposta fornecida pelo IFBot (Figura 22). Isso é particularmente importante, pois permite que os usuários expressem seu nível de satisfação com as respostas recebidas. Assim, caso o usuário considere uma resposta insatisfatória ou até mesmo incorreta, ele pode avaliá-la negativamente, fornecendo um *feedback* instantâneo que fica registrado para que os administradores do *chatbot* consigam melhorá-lo continuamente.



Figura 20. Tela inicial do IFBot. Fonte: Autora

Conforme pode ser observado, a Figura 21 apresenta a tela de *chat* versão *web*, que é onde ocorre a interação do usuário com o *chatbot*. Nesta tela, os usuários podem iniciar uma conversa com o IFBot. Uma característica conveniente é a presença da barra lateral que oferece opções de tópicos pré-definidos, permitindo aos usuários a escolha de um assunto específico para iniciar a conversa de forma mais prática. Isso não apenas simplifica a experiência, mas também agiliza o acesso a informações acadêmicas básicas (renovação de matrícula, trancamento, aproveitamento, salas de aula, horário, etc).

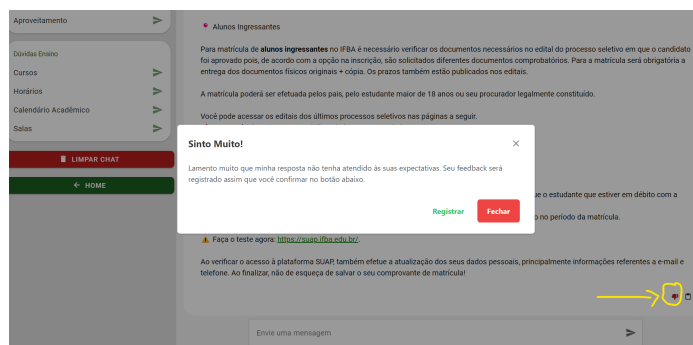


Figura 22. Registro de *feedback* negativo para uma mensagem do IFBot. Fonte: Autora

Além da interação através do *chat*, a aplicação também oferece um painel administrativo. O acesso a este painel é restrito e requer autenticação com um nome de usuário e senha. Após a autenticação bem-sucedida, o usuário é redirecionado para a tela principal do painel (Figura 23). Neste ambiente, os administradores têm acesso a uma série de funcionalidades essenciais para o gerenciamento do IFBot.

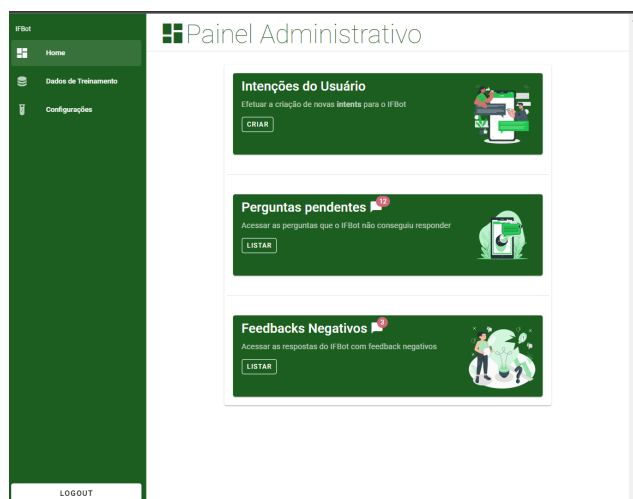


Figura 23. Tela principal do painel administrativo do IFBot. Fonte: Autora

Conforme pode ser observado na Figura 23, na tela principal do painel administrativo, pode-se encontrar opções para criar novas *intents*, listar as perguntas não respondidas pelo IFBot e listar as mensagens que receberam *feedback* negativo dos usuários.

A tela de Criação de *Intents* representa um componente fundamental para o sistema. Através desta é possível expandir e aprimorar continuamente a inteligência do *bot*, permitindo a criação de novas intenções de usuários. Tal funcionalidade traz versatilidade ao IFBot, pois a medida que novos contextos e intenções dos usuários surgem, os administradores podem incorporá-los rapidamente ao sistema, permitindo que o *bot* evolua constantemente para atender às demandas e conseguir compreender e responder a novas perguntas dos estudantes do IFBA - Campus Salvador.

Caso o IFBot não consiga responder a alguma pergunta do usuário, ele armazenará o conteúdo desta mensagem e a data da ocorrência. Tais dados ficarão disponíveis através da tela de Mensagens Não Respondidas. Caso a pergunta for relevante, o administrador poderá criar uma nova *intent* ou incluir essa frase em alguma *intent* já existente e, assim, ampliar a capacidade de atendimento do *bot* de acordo com as demandas apresentadas pela comunidade do IFBA. Caso a mensagem não tenha nenhuma relevância o administrador pode apenas “Concluir-la” e ela não será mais listada.

Se o usuário considerou a resposta do IFBot insatisfatória, tais registros ficarão disponíveis para serem visualizados pelo administrador através da tela de *Feedbacks* Negativos. Com isso, é possível identificar áreas problemáticas do IFBot e resolvê-las através da criação de uma nova intenção, da inclusão da pergunta em alguma intenção já existente ou ajustar a resposta do *bot* caso contenha algum erro na mesma.

A tela de Dados de Treinamento apresenta a listagem de todas as *intents* cadastradas no IFBot, além de também ser possível realizar alterações que se mostrem necessárias. Essa tela também conta com filtros pela categoria e *tag* das intenções, que permite uma busca mais ágil e eficiente.

Por fim, a cada alteração realizada no *bot*, é necessário que este seja novamente treinado para que as modificações sejam

efetuadas. Por meio da tela configuração, o administrador poderá realizar o treinamento do IFBot. Além disso, ainda nesta tela é possível efetuar a integração com a planilha alocação de salas por meio da URL da mesma, para que o IFBot consiga ler essa planilha e informar sobre as localizações de salas de aulas das disciplinas.

As imagens das telas que não foram exibidas na presente seção e suas descrições encontram-se na seção VII deste artigo, assim como o diagrama de casos de uso do sistema.

## V. AVALIAÇÕES E RESULTADOS

Com o intuito de avaliar a solução proposta, foram feitos testes com usuários, visando obter uma percepção inicial sobre o eficácia e a utilização do IFBot.

Os testes de conversação foram realizados por estudantes do IFBA de diferentes modalidades de ensino, entre os dias 7 a 15 de março de 2024. O propósito foi avaliar não apenas a capacidade do IFBot em fornecer informações acadêmicas relevantes sobre o IFBA - Campus Salvador, como também a experiência do usuário durante a interação com o *bot*.

Inicialmente foi criado um documento no *Google Docs*, disponível através do link <https://docs.google.com/document/d/14Z64dB3N>, que continha instruções sobre como acessar o sistema, além de descrever os temas que englobam a atual base de conhecimento do IFBot (informações de matrícula, ensino, cursos, localizações de salas de aula, entre muitos outros). Posteriormente esse documento foi compartilhado com os participantes e foi solicitado que fossem feitas perguntas para o *bot* dentro destes temas. No documento disponibilizado, juntamente com as instruções havia o *link* de uma pesquisa *online*, feita através do *Google Forms*, para ser respondida ao final do teste.

No total, 38 usuários responderam a pesquisa. A seguir encontram-se as questões propostas e as respostas obtidas.

Em relação a modalidade de ensino do IFBA, a pesquisa contou com 35 **Estudantes de Graduação**, 2 **Estudantes do Ensino Técnico** e 1 **Estudante da Pós-graduação**, conforme pode ser observado na Figura 24.

Qual sua modalidade de ensino no IFBA?

38 respostas

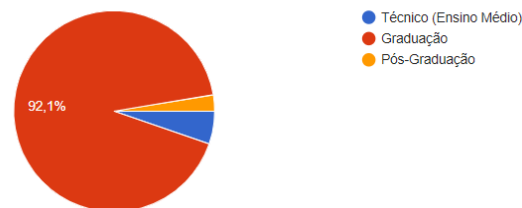


Figura 24. Pergunta sobre modalidade de ensino do estudante. Fonte: Autora

Sobre a utilidade do sistema para ter um acesso mais rápido a informações acadêmicas do IFBA, 28 pessoas avaliaram como **Muito útil** e 10 pessoas avaliaram como **Útil**, conforme pode ser observado na Figura 25.

### Quão útil você achou o IFBot para ter um acesso mais rápido a informações acadêmicas do IFBA?

38 respostas

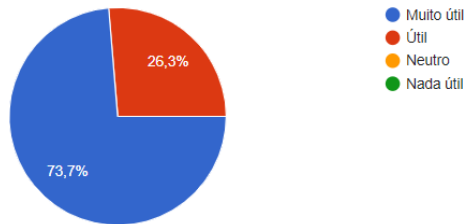


Figura 25. Pergunta sobre utilidade do sistema. Fonte: Autora

Os usuários foram questionados sobre quais os problemas o IFBot poderia resolver ou quais facilidades ele poderia trazer para os estudantes do IFBA. Algumas respostas que foram coletadas encontram-se a seguir:

- “Facilitar o acesso a informações centralizando as diversas fontes oficiais do IFBA em uma só ferramenta”;
- “Evitar aberturas de chamados triviais, descongestionando o pessoal para que possa atender chamados mais prioritários”;
- “Ele traz a facilidade de informações rápidas com uma linguagem acessível ao público geral, já que ler os documentos oficiais é a última opção para a maioria das pessoas, geralmente perdemos tempo perguntando a colegas, professores ou mandando *e-mails* para GRA ou Coordenação do curso”;
- “Desburocratizar o acesso a informações que hoje só estão disponíveis no atendimento da coordenação”;
- “O IFBot deixa muito mais prático a busca de informações que normalmente estão espalhadas pelo site oficial do Ifba e documentos”;
- “Atualmente, toda e qualquer informação relacionada ao IFBA está muito descentralizada e disposta de maneira confusa/bagunçada, o IFBot tem o poder de centralizar essas informações e servi-las de forma eficiente e facilitada”;
- “Informações ágeis e rápidas. Quando a administração do curso demora para responder um *e-mail*, ele pode dar a resposta em curto espaço de tempo ajudando bastante”;
- “Agilidade e facilidade de acesso as informações já que reúne vários segmentos em um lugar único de pesquisa, desde onde encontrar as salas de aula a informações de procedimentos do campus”;
- “Informações básicas que não são fornecidas precisamente e de fácil acesso aos alunos”;
- “Maior facilidade de informações e velocidade de resposta, pode ajudar muito em casos onde não saber onde perguntar ou acabar ficando perdido de alguma forma”;

- “Acesso a documentos específicos informativos sobre curso; encontrar a sala correspondente da disciplina tal como obter informações acerca do próprio acesso aos endereços associados”;
- “Instruções acerca de documentação e matrícula, essencial para aqueles que cogitam entrar na instituição ou acabou de o fazer e etc... Parece uma ferramenta útil e conveniente em termos de reunir documentos, endereços e informações de utilidade geral voltada ao IFBA”;
- “Achar informações no site pode ser desafiador no primeiro momento, imagino que o *Bot* possa servir como intermediário para facilitar o acesso mais direto ao conteúdo”.

Quando questionados sobre a facilidade de uso do sistema, 28 pessoas responderam que foi **Muito Fácil** e 10 responderam que foi **Fácil**, conforme pode ser observado na Figura 26.

### Quão fácil foi para você usar o IFBot?

38 respostas

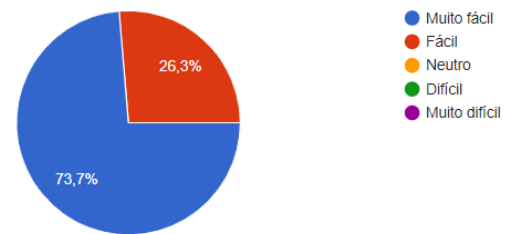


Figura 26. Pergunta sobre facilidade de uso do sistema. Fonte: Autora

Em relação à frequência com que o IFBot conseguiu responder às perguntas feitas, 16 pessoas o avaliaram como **Sim, sempre**, 20 responderam **Na maioria das vezes** e 2 avaliaram como **Às vezes**, conforme demonstrado na Figura 27.

### Dentro dos temas que foram sugeridos, o IFBot conseguiu responder suas perguntas?

38 respostas

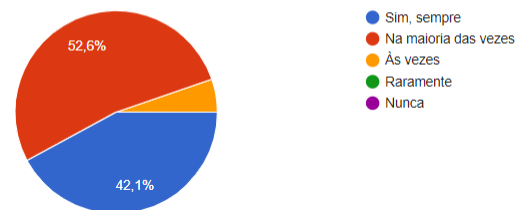


Figura 27. Pergunta sobre capacidade de resposta do IFBot. Fonte: Autora

Os participantes foram solicitados a classificar o sistema com base na experiência obtida, usando uma escala de 1 a 5. 11 pessoas avaliaram com nota 4 e 27 pessoas com nota 5, conforme mostrado na Figura 28.

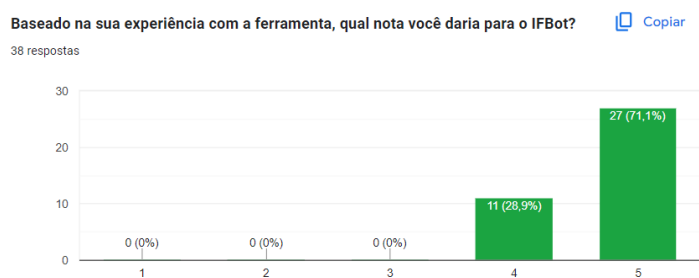


Figura 28. Pergunta sobre nota do IFBot. Fonte: Autora

Por último, foi solicitado um *feedback* sobre melhorias e futuras funcionalidades que poderiam ser implementadas no IFBot. Algumas das respostas obtidas estão listadas a seguir:

- “Integração com SUAP futuramente”;
- “Menos *links*, mais respostas”;
- “Eu acho que poderia colocar imagens também, por exemplo, a pessoa quer saber onde fica o bloco O, o *bot* informaria onde fica, juntamente com uma foto”;
- “Sem sugestões específicas. Apenas continuar acrescentando mais informações que o IFBot ainda não tem conhecimento”;
- “Aumentar a compreensão do IFBot do que está sendo pedido e a melhorar a capacidade do IFBot de responder diretamente, ao invés de redirecionar o usuário para uma outra página”;
- “Deixo como sugestão a possibilidade futura de um *app* que torne o processo de encontrar informações pertinentes ao IFBA ainda mais fácil e prático!”;
- “Podemos abranger todo o conhecimento do IFbot, de que forma? Como não tem uma IA conectada aqui, sugiro fazer um levantamento com os alunos das dificuldades e principais dúvidas que os estudantes tem atualmente. Por exemplo: comecei a estagiar em novembro, não entreguei o documento para o IFBA assinar de imediato e acabei perdendo o tempo corrido, mas esse detalhe não é informado assim sabe... As principais dúvidas podem ser filtradas por uma enquete para os alunos. E deixar sempre sugestões para os alunos questionarem... e assim o IFbot filtrar essas perguntas de maneira mais real a necessidades”;
- “Sistema para fazer uma rolagem de conversa como um *chat*”.

## VI. CONCLUSÃO

Dentre as transformações e adversidades trazidas pela era digital destaca-se a crescente demanda por informações relevantes em meio a um volume exponencial de dados. No presente trabalho foi abordado os desafios associados a busca por informações acadêmicas no Instituto Federal da Bahia (IFBA) - Campus Salvador, e como isto está correlacionado a quantidade de dados massiva distribuídas por inúmeras

páginas e documentos dos portais web da instituição. Neste cenário, é muito comum que os discentes precisem recorrer a diversos *e-mails*, coordenações e outros setores administrativos e pedagógicos para esclarecer dúvidas, que muitas vezes são extremamente simples, pois a instituição não conta com um canal de comunicação unificado que atenda aos diferentes setores presentes no campus.

É nesse contexto que ferramentas que auxiliem na filtragem e no acesso à informação tornam-se extremamente necessárias. Com as evoluções ocorridas no campo da inteligência artificial, principalmente nas áreas de *machine learning* e processamento de linguagem natural, surgiram os *chatbots*, que passaram a suprir tal demanda e são aplicados nos mais variados campos como educação, negócios e comércio eletrônico, saúde e entretenimento [5]. Os *chatbots* são capazes de oferecer aos usuários uma assistência confortável e eficiente, além de serem capazes de atender múltiplas pessoas simultaneamente [5].

Deste modo, este trabalho foi desenvolvido com o objetivo de criar-se o IFBot, um *chatbot* informacional para o Instituto Federal da Bahia - Campus Salvador, visando agilizar e otimizar a busca por informações do ambiente acadêmico por parte dos estudantes, permitindo que estes obtenham respostas relevantes em tempo real.

Para construção do IFBot, utilizou-se a linguagem de programação *Python*, o *Spacy* no pre-processamento das informações, o *Tensorflow* e a *API Keras* para a construção da rede neural e treinamento do *bot*. Para a elaboração da interface *web* foram utilizados a *Vue.js* e para o armazenamento dos dados foi utilizado o *MongoDB*.

Através da análise e dos resultados dos testes de conversação realizados com estudantes do IFBA, evidencia-se a utilidade da ferramenta para o acesso rápido a informações acadêmicas da instituição, a maioria dos participantes da pesquisa efetuada o considerou muito útil. Os *feedbacks* coletados destacam a capacidade do IFBot em centralizar informações dispersas e proporcionar respostas ágeis e acessíveis.

Além disso, é importante destacar o papel do IFBot como uma ferramenta de inovação e transformação digital na educação, demonstrando o potencial das tecnologias de *chatbots* na melhoria da experiência do aluno.

### A. Limitações deste Trabalho

O presente trabalho apresenta algumas limitações, dentre estas pode-se citar:

- 1) Devido às restrições de tempo e à diversidade das informações disponíveis nos portais *web* da instituição, não foi possível criar uma base de dados extremamente completa com informações sobre o Campus Salvador. Assim, o escopo de conhecimento do IFBot pode não ter contemplado todas as necessidades informacionais dos estudantes do IFBA. Desse modo, o *bot* muitas vezes apresenta algumas respostas mais genéricas, como também efetua o redirecionamento do usuário para o site da instituição para obter o dado desejado;
- 2) Devido a complexidade da ferramenta, não foi viável hospedar o sistema em um servidor gratuito, o que, devido aos custos, resultou na impossibilidade

de disponibiliza-lo de forma permanente para toda comunidade acadêmica do IFBA - Campus Salvador.

## B. Trabalhos Futuros

Levando-se em consideração os *feedbacks* obtidos através das pesquisas realizadas com os usuários, alguns novos recursos podem trazer muitos benefícios para a evolução do IFBot. Entre eles destacam-se:

- 1) Expandir a base de conhecimentos do IFBot, visando fornecer respostas mais específicas e mais completas, com menos redirecionamento para o site da instituição;
- 2) Melhorias na interface tais como: otimizações na rolagem do *chat*, ajustes de responsividade no modo *mobile*;
- 3) Implementação da inclusão de imagens no *chat*;
- 4) Implementação da possibilidade de salvar o histórico do *chat*;
- 5) Implementação do envio de um *feedback* escrito sobre a utilização do IFBot;
- 6) Implementação de recursos para gerenciar o perfil do usuário no painel administrativo;
- 7) Integração com o SUAP.

## REFERÊNCIAS

- [1] B. P. BELÉN. (2023) A era dos dados: quantos geramos e como isso impacta nossa vida. [Online]. Available: <https://linkages.com.br/2023/03/29/dados-quantos-geramos-e-como-isso-impacta-nossa-vida/>
- [2] L. B. MELO, “Regbot: Chatbot para o regulamento de ensino de graduação da universidade federal de campina grande,” p. 16, 2023. [Online]. Available: <http://dSPACE.sti.ufcg.edu.br:8080/jspui/handle/riufcg/30491>
- [3] M. TIME. (2023) Pesquisa panorama mobile time - mapa do ecossistema brasileiro de bots - agosto de 2023. [Online]. Available: <https://www.mobiletime.com.br/pesquisas/>
- [4] D. LÉXICO. (2023) Definição de chatbot em inglês. [Online]. Available: <https://www.lexico.com/en/definition/chatbot>
- [5] E. ADAMOPOULOU and L. MOUSSIADES, “Chatbots: History, technology, and applications,” *Machine Learning with Applications*, vol. 2, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666827020300062>
- [6] P. A. SABAINI, B. G. CLEMENTE, and B. C. COUTINHO, “Proposta de assistente virtual inteligente para estudantes das instituições federais de ensino profissional,” *Instituto Federal do Espírito Santo Campus Serra*, 2023. [Online]. Available: <https://sol.sbc.org.br/index.php/encompif/article/view/11069/10940>
- [7] K. K. C. SILVA and et al., “Desenvolvimento de ferramenta de chatbot como solução para a comunicação do ifb,” *Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Campus Brasília*, 2023. [Online]. Available: <https://sol.sbc.org.br/index.php/encompif/article/view/11069/10940>
- [8] M. H. SILVA, “Chatbot como ferramenta para facilitar o acesso a informações das instituições federais de ensino: um estudo de caso no instituto federal do espírito santo - campus cachoeiro de itapemirim,” p. 58, 2022. [Online]. Available: <https://repositorio.ifes.edu.br/handle/123456789/2677?show=full>
- [9] C. W. OKONKWO and A. ADE-IBIJOLA, “Chatbots applications in education: A systematic review,” *Computers and Education: Artificial Intelligence*, vol. 2, no. 100033, p. 100033, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666920X21000278>
- [10] S. RUSSELL and P. NORVIG, *Inteligência artificial*, 3rd ed. Elsevier, 2013.
- [11] C. MANNING. (2023) Artificial intelligence definitions. [Online]. Available: <https://hai.stanford.edu/sites/default/files/2020-09/AI-Definitions-HAI.pdf>
- [12] D. D. GOMES. (2023) Inteligência artificial: Conceitos e aplicações. [Online]. Available: [https://www.professores.uff.br/screspo/wp-content/uploads/sites/127/2017/09/ia\\_intro.pdf](https://www.professores.uff.br/screspo/wp-content/uploads/sites/127/2017/09/ia_intro.pdf)
- [13] X. C. BARBOSA and R. F. BEZERRA, “Breve introdução à história da inteligência artificial,” *Jamaxi, UFAC*, vol. 4, no. 2, 2020.
- [14] S. L. PEREIRA. (2023) Introdução à inteligência artificial. [Online]. Available: <https://www.ime.usp.br/~slago/IA-introducao.pdf>
- [15] P. ONLINE. (2023) Inteligência artificial: O que é e como funciona. [Online]. Available: <https://online.pucrs.br/blog/inteligencia-artificial>
- [16] R. ANYOHA. (2023) The history of artificial intelligence. [Online]. Available: <https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/>
- [17] E. ESCOTT. (2023) What are the 3 types of ai? a guide to narrow, general, and super artificial intelligence. [Online]. Available: <https://codebots.com/artificial-intelligence/the-3-types-of-ai-is-the-third-even-possible>
- [18] T. B. LUDERMIR, “Inteligência artificial e aprendizado de máquina: estado atual e tendências,” *Estudos Avançados*, vol. 35, no. 101, pp. 85–94, 2021.
- [19] J. CARBONELL and et al., *Machine Learning: An Artificial Intelligence Approach*. Palo Alto, California: TIOGA Publishing Co., 1983.
- [20] D. L. SILVA, “Lia: um chatbot inteligente para o domínio de imóveis,” p. 79, 2019.
- [21] M. C. MONARD and J. A. BARANAUSKAS. Conceitos sobre aprendizado de máquina. [Online]. Available: <https://dcm.ffclrp.usp.br/~agosto/publications/2003-sistemas-inteligentes-cap4.pdf>
- [22] B. M. SILVA and M. VANDERLINDE. Inteligência artificial, aprendizado de máquina. [Online]. Available: [https://www.ceavi.udesc.br/arquivos/id\\_submenu/387/brigiane\\_machado\\_da\\_silva\\_\\_marcos\\_vanderlinde.pdf](https://www.ceavi.udesc.br/arquivos/id_submenu/387/brigiane_machado_da_silva__marcos_vanderlinde.pdf)
- [23] C. JANIESCH, P. ZSCHECH, and K. HEINRICH, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, p. 685–695, 2021.
- [24] G. E. de Almeida Prado Alves Batista, “Pré-processamento de dados em aprendizado de máquina supervisionado,” p. 232, 2003. [Online]. Available: <https://www.teses.usp.br/teses/disponiveis/55/55134/tde-06102003-160219/publico/TeseDoutorado.pdf>
- [25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive Computation and Machine Learning series. MIT Press, 2016. [Online]. Available: <https://books.google.com.br/books?id=omivDQAAQBAJ>
- [26] J. Kelleher, *Deep Learning*, ser. The MIT Press Essential Knowledge series. MIT Press, 2019. [Online]. Available: <https://books.google.com.br/books?id=b06qDwAAQBAJ>
- [27] M. Ekman, *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, NLP, and Transformers Using Tensorflow*, 1st ed. Addison-Wesley Professional, 2021, accessed October 9, 2023. [Online]. Available: <https://www.vlebooks.com/vleweb/product/openreader?id=none&isbn=9780137470297>
- [28] D. Kriesel. (2005) A brief introduction to neural networks. [Online]. Available: [https://www.dkriesel.com/\\_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf](https://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf)
- [29] I. Data and A. Team. (2023) Ai vs. machine learning vs. deep learning vs. neural networks: What’s the difference? [Online]. Available: <https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>
- [30] L. Hardesty. (2017) Explained: Neural networks. [Online]. Available: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>
- [31] G. Boesch. (2024) Deep neural network: The 3 popular types (mlp, cnn and rnn). [Online]. Available: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types/>
- [32] A. Chopra, A. Prashar, and C. Sain. (2023) Natural language processing. [Online]. Available: [https://citeseerx.ist.psu.edu/document?repid=\\$rep1&type=\\$pdf&doi=\\$%\\$eace1d14e266a5cd44fe781a874c662928602fd](https://citeseerx.ist.psu.edu/document?repid=$rep1&type=$pdf&doi=$%$eace1d14e266a5cd44fe781a874c662928602fd)
- [33] D. KHURANA and et al., “Natural language processing: state of the art, current trends and challenges,” *Multimedia tools and applications*, vol. 82, no. 3, p. 3713–3744, 2023.

- [34] E. Liddy, *Natural Language Processing*, 2nd ed. NY: Marcel Decker, Inc., 2001.
- [35] C. N. Aranha, "Processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional," Rio de Janeiro, 2007.
- [36] OpenAI. (2022) Introducing chatgpt. [Online]. Available: <https://openai.com/blog/chatgpt>
- [37] P. Singh, "Chatbot intent - types, classifications, and training steps," *REVE Chat*. [Online]. Available: <https://www.revechat.com/blog/chatbot-intents/>
- [38] J. Lennon, *Beginning CouchDB*. Berkeley, CA: Apress, 2009.
- [39] G. Zhao, W. Huang, S. Liang, and Y. Tang, "Modeling mongodb with relational model," *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, pp. 115–121, 2013. [Online]. Available: <https://doi.org/10.1109/EIDWT.2013.25>
- [40] MongoDB. Customer success stories. [Online]. Available: <https://www.mongodb.com/who-uses-mongodb>
- [41] —. Collections. [Online]. Available: <https://www.mongodb.com/docs/compass/current/collections>
- [42] (2023) How spacy works. [Online]. Available: <https://explosion.ai/blog/how-spacy-works>
- [43] SpaCy. (2023) Industrial-strength natural language processing (nlp) in python. [Online]. Available: <https://explosion.ai/blog/how-spacy-works>
- [44] (2023) Portuguese · spacy models documentation. [Online]. Available: <https://spacy.io/models/pt>
- [45] J. Brownlee, *Deep Learning With Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*. Machine Learning Mastery, 2016. [Online]. Available: <https://books.google.com.br/books?id=K-ipDwAAQBAJ>
- [46] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [47] A. P. Braga, A. P. L. F. Carvalho, and T. B. Ludemir, *Fundamentos de Redes Neurais Artificiais*. Rio de Janeiro: Atas da 11ª Escola de Computação, 1998.
- [48] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [49] Awari. (2023) Dropout no tensorflow: Melhorando a regularização de modelos de aprendizado de máquina. [Online]. Available: <https://awari.com.br/dropout-no-tensorflow-melhorando-a-regularizacao-de-modelos-de-aprendiz>
- [50] FastAPI. (2024) Fastapi. [Online]. Available: <https://fastapi.tiangolo.com/>
- [51] IBM. (2024) Json web token (jwt). [Online]. Available: <https://www.ibm.com/docs/pt-br/cics-ts/6.1?topic=cics-json-web-token-jwt>
- [52] Awari. (2023) Bcrypt python: Aprenda a criptografar senhas com python. [Online]. Available: <https://awari.com.br/bcrypt-python-aprenda-a-criptografar-senhas-com-python>
- [53] Vue.js. (2023) Introdução: O que é vue.js. [Online]. Available: <https://br.vuejs.org/v2/guide/index.html>
- [54] Vuetify. Introduction. [Online]. Available: <https://vuetifyjs.com/en/introduction/why-vuetify/>
- [55] O. Filipova, *Learning Vue.js 2*. Packt Publishing, 2016. [Online]. Available: <https://books.google.com.br/books?id=nszcDgAAQBAJ>
- [56] I. Sommerville, *Software engineering*, 9th ed. Pearson Education, 2016.
- [57] Amazon. What is aws? [Online]. Available: [https://aws.amazon.com/pt/what-is-aws/?nc1=f\\_cc](https://aws.amazon.com/pt/what-is-aws/?nc1=f_cc)
- [58] —. Amazon ec2. [Online]. Available: <https://aws.amazon.com/pt/ec2>

## VII. APÊNDICE A

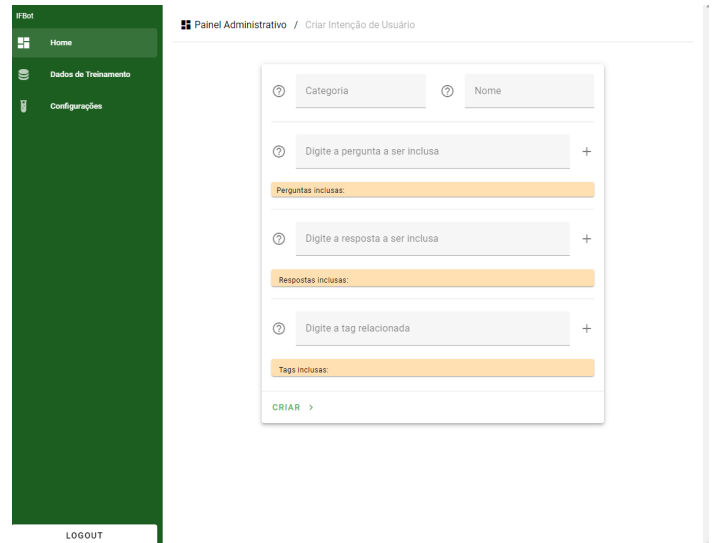


Figura 29. Tela de cadastro de intenções de usuário. Fonte: Autora

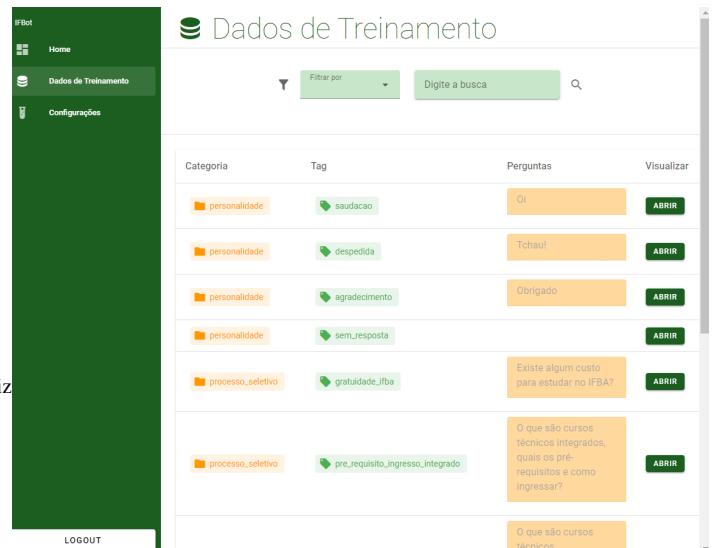


Figura 30. Tela de listagem dos dados de treinamento (*intents*). Fonte: Autora

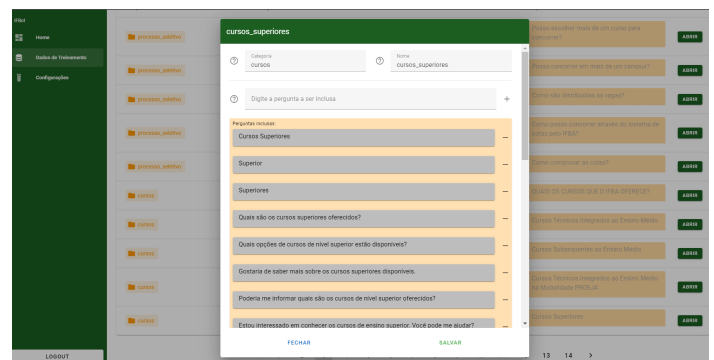


Figura 31. Tela de edição de uma intenção de usuário. Fonte: Autora

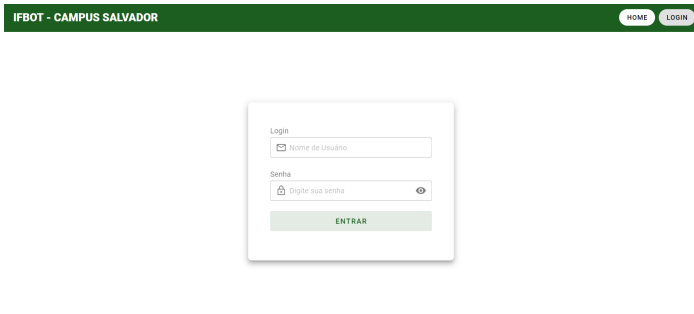


Figura 32. Tela de login. Fonte: Autora

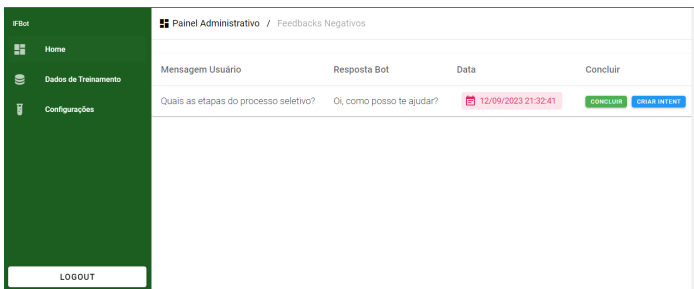


Figura 33. Tela de listagem de feedbacks negativos. Fonte: Autora

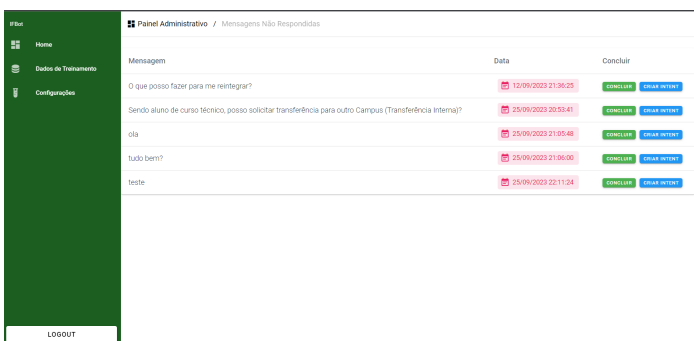


Figura 34. Tela de listagem de perguntas não respondidas pelo IFBot. Fonte: Autora

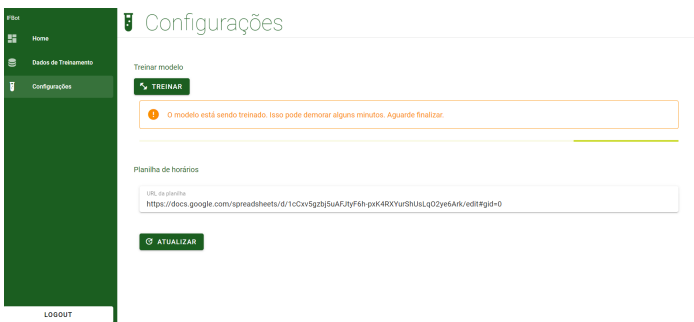


Figura 35. Tela de Configuração do IFBot. Fonte: Autora

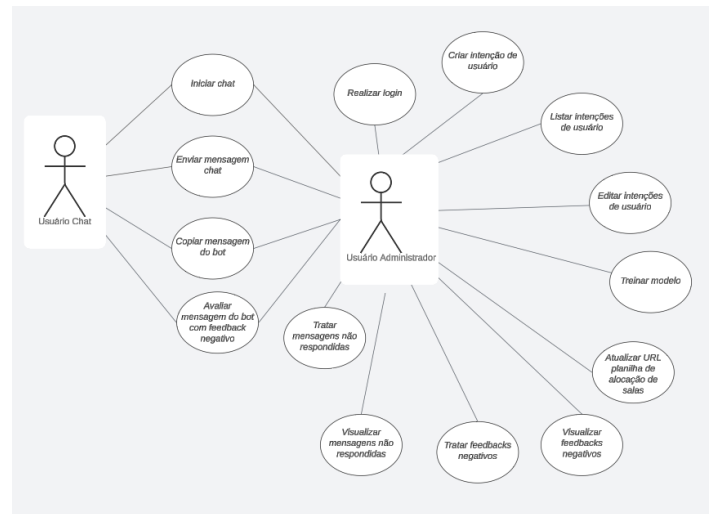


Figura 36. Diagrama de casos de uso do sistema. Fonte: Autora