PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DESENVOLVIMENTO DE UM CHATBOT INTELIGENTE PARA O ATENDIMENTO AO CLIENTE

JOSÉ ENRIQUE BALIEIRO QUARESMA

GOIÂNIA-GO
2025

JOSÉ ENRIQUE BALIEIRO QUARESMA

DESENVOLVIMENTO DE UM CHATBOT INTELIGENTE PARA O ATENDIMENTO AO CLIENTE

Trabalho de conclusão de curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Fernando Gonçalves Abadia

GOIÂNIA-GO

2025

JOSÉ ENRIQUE BALIEIRO QUARESMA

DESENVOLVIMENTO DE UM CHATBOT INTELIGENTE PARA O ATENDIMENTO AO CLIENTE

Este Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica e Artes, da Pontifícia Universidade Católica de Goiás em 06/06/ 2025.

_____

Orientador: Prof. Me. Fernando Gonçalves Abadia

_____

Prof. Rafael Leal Martins

_____

Prof. Anibal Santos Jukemura

GOIÂNIA-GO

2025

RESUMO

Este Trabalho de Conclusão de Curso tem como objetivo desenvolver um protótipo de chatbot inteligente voltado para o atendimento automatizado ao cliente, utilizando técnicas de Processamento de Linguagem Natural (PLN). A motivação parte da crescente demanda por soluções automatizadas que otimizem o tempo de resposta, aumentem a eficiência operacional e melhorem a experiência do usuário. A fundamentação teórica aborda os conceitos de Inteligência Artificial, evolução dos chatbots, modelos de PLN, arquitetura de sistemas inteligentes e tecnologias utilizadas. A metodologia aplicada consiste em uma pesquisa aplicada e exploratória, com desenvolvimento prático em um ambiente de testes utilizando frameworks e bibliotecas amplamente utilizados na área, como Flask, spaCy e ngrok. O protótipo desenvolvido é capaz de identificar intenções básicas dos usuários, responder de forma automatizada e registrar os dados das interações. Apesar de suas limitações, como a simplicidade das respostas, o sistema demonstrou potencial para aplicações reais. Por fim, são apresentadas sugestões de melhorias futuras, incluindo o aprimoramento do modelo de linguagem e a integração com plataformas externas.

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

ABSTRACT

This undergraduate thesis aims to develop an intelligent chatbot prototype for automated customer service, using Natural Language Processing (NLP) techniques. The motivation stems from the increasing demand for automated solutions that optimize response time, improve operational efficiency, and enhance user experience. The theoretical foundation addresses concepts of Artificial Intelligence, chatbot evolution, NLP models, intelligent system architecture, and the technologies used. The applied methodology consists of an exploratory and applied research approach, with practical development in a test environment using widely adopted frameworks and libraries such as Flask, spaCy, and ngrok. The developed prototype is capable of identifying basic user intents, responding automatically, and recording interaction data. Despite limitations, such as the simplicity of its responses, the system showed potential for real-world applications. Finally, suggestions for future improvements are presented, including the enhancement of the language model and integration with external platforms.

Keywords: Chatbot, Artificial Intelligence, Natural Language Processing.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

O avanço da Inteligência Artificial (IA) tem transformado diversas áreas, incluindo o atendimento ao cliente. Um dos principais desenvolvimentos nessa área é o chatbot, um software projetado para simular interações humanas por através de mensagens de texto ou voz. Chatbots utilizam algoritmos de aprendizado de máquina e Processamento de Linguagem Natural (PLN) para responder a perguntas e realizar tarefas de forma eficiente e automatizada (INÁCIO, 2024). Esses sistemas já se tornaram populares em empresas de diferentes setores, como bancos, e-commerce e telecomunicações, devido à sua capacidade de melhorar a experiência do usuário e reduzir custos operacionais.

Em 2022 houve um grande crescimento no uso de chatbots inteligentes. Segundo estudo da empresa de Markets and Markets (2023), o mercado global de chatbots está em constante expansão, com previsões de crescimento anual de 23.3% até 2028. A adoção de chatbots inteligentes para atendimento ao cliente tem sido uma tendência predominante, especialmente com o uso de IA e PNL cada vez mais sofisticados, que permitem respostas mais precisas e personalizadas. Além disso, conforme Melo (2019), a integração desses sistemas com bancos de dados corporativos e CRM (Customer Relationship Management) tem potencializado sua eficácia, permitindo um atendimento mais ágil e adaptado às necessidades do cliente.

Estudar o desenvolvimento de um chatbot inteligente para atendimento ao cliente é relevante, pois essa tecnologia representa uma solução eficiente para empresas que buscam melhorar a qualidade de seu atendimento, aumentar a satisfação do cliente e otimizar seus recursos internos. Além disso, com o aumento da demanda por soluções automatizadas e interativas, há uma necessidade crescente de explorar como o uso de IA pode ser aplicado de forma mais eficaz em ambientes corporativos (MELLO FILHO, 2024). Portanto, investigar o desenvolvimento e implementação de chatbots inteligentes pode gerar insights valiosos para organizações que desejam aprimorar seus processos de atendimento ao cliente.

Diante desse contexto, este projeto busca responder à seguinte questão: **Como desenvolver um chatbot inteligente que otimize o atendimento ao**

cliente, proporcionando respostas precisas e melhorando a experiência do usuário?

1.1 Objetivos

Para o entendimento desse trabalho segue os objetivos abaixo:

1.1.1 Geral

Desenvolver um protótipo de chatbot inteligente, utilizando algoritmos de PLN, voltado para o atendimento automatizado ao cliente.

1.1.2 Específicos

- Pesquisar as principais tecnologias e frameworks de IA e PLN utilizados no desenvolvimento de chatbots;
- Projetar a arquitetura de um chatbot para simular interações comuns em um cenário de atendimento ao cliente;
- Implementar o protótipo do chatbot com foco em respostas automatizadas e interação básica com o usuário;
- Documentar o processo de desenvolvimento e resultados obtidos no protótipo, propondo sugestões para futuras melhorias

1.2 Justificativa

A crescente digitalização dos serviços e a demanda por atendimentos mais ágeis, personalizados e contínuos impulsionaram a adoção de soluções baseadas em IA no relacionamento com o cliente. Nesse cenário, os chatbots inteligentes se destacam como ferramentas versáteis e eficazes, capazes de proporcionar experiências mais dinâmicas, reduzir o tempo de resposta e operar 24 horas por dia (SILVA, 2024).

Além disso, o uso de algoritmos de PLN permite que esses sistemas compreendam melhor as intenções dos usuários, respondam de maneira mais natural e simulem conversas humanas com mais eficiência. Com o crescimento

do interesse por essa tecnologia em ambientes corporativos, torna-se pertinente investigar sua aplicação prática mesmo em soluções de menor complexidade (SILVA, 2024).

Assim, este projeto se justifica pela relevância acadêmica e tecnológica do tema, ao reunir fundamentos de IA e PLN em um contexto de atendimento automatizado. A proposta de desenvolver um protótipo simples, voltado para interações básicas e registro de dados, tem como objetivo explorar a viabilidade da aplicação desses conceitos e servir de base para aprimoramentos e estudos futuros na área.

1.3 Estrutura do trabalho

Diante da relevância do tema e dos objetivos propostos, este Trabalho de Conclusão de Curso (TCC) está estruturado da seguinte forma:

O Capítulo 1 apresenta a introdução ao tema, contextualizando o uso da Inteligência Artificial no atendimento ao cliente, bem como os objetivos, a justificativa e a estrutura geral do trabalho; O Capítulo 2 aborda a fundamentação teórica, incluindo os conceitos e a evolução da Inteligência Artificial, os tipos e aplicações de chatbots, os fundamentos do PLN, a arquitetura de sistemas inteligentes e as principais tecnologias e ferramentas utilizadas no desenvolvimento desses sistemas; O Capítulo 3 descreve os materiais e métodos empregados, destacando o tipo de pesquisa adotado, o ambiente de desenvolvimento e os recursos utilizados durante a construção do chatbot; O Capítulo 4 apresenta os resultados obtidos, detalhando as etapas do desenvolvimento do chatbot, a seleção das tecnologias e ferramentas aplicadas, bem como a modelagem da arquitetura do sistema; O Capítulo 5 é dedicado à estrutura do protótipo desenvolvido, demonstrando seu funcionamento prático, as limitações observadas durante os testes e as sugestões de melhorias; Por fim, o Capítulo 6 reúne as considerações finais do trabalho, refletindo sobre os principais resultados alcançados e propondo direções para pesquisas e desenvolvimentos futuros na área.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo reúne os principais conceitos, definições, tecnologias e abordagens relacionados ao desenvolvimento de chatbots inteligentes com base em IA e PLN. Serão abordadas as origens e aplicações da IA, os diferentes tipos e a evolução dos chatbots, os fundamentos e modelos de PLN, a estrutura técnica necessária para implementação de um chatbot e, por fim, as ferramentas e tecnologias mais utilizadas na área.

2.1 Inteligência Artificial e suas Aplicações

A Inteligência Artificial é um campo da ciência da computação que busca desenvolver sistemas capazes de simular a inteligência humana. Seu estudo é fundamental para a construção de sistemas como chatbots, que precisam compreender, processar e reagir a estímulos de forma autônoma (FERREIRA, 2024). Nesta seção, serão abordados os conceitos básicos da IA, sua evolução ao longo do tempo e as aplicações práticas no setor de serviços, especialmente no atendimento ao cliente

2.1.1 Conceitos e Evolução da IA

A IA tem ganhado destaque em diversos setores, incluindo a educação, onde vem sendo aplicada para otimizar processos e apoiar práticas pedagógicas, conforme discutido por Barbosa e Portes (2023). Seu conceito surgiu formalmente em 1956, durante a Conferência de Dartmouth, quando pesquisadores como John McCarthy, Marvin Minsky e Alan Newell propuseram que a inteligência humana poderia ser descrita de maneira tão precisa que uma máquina poderia simulá-la. Desde então, a IA passou por diversas fases, inicialmente com grande otimismo e desenvolvimento de sistemas baseados em regras e lógica simbólica, conhecidos como “IA simbólica” ou “GOFAI” (Good Old-Fashioned Artificial Intelligence). No entanto, as limitações desses sistemas, principalmente no que tange à escalabilidade e à adaptação a contextos dinâmicos, resultaram em períodos de estagnação conhecidos como “invernos da IA” (FERREIRA, 2024).

A partir dos anos 2000, com o avanço do poder computacional, a disponibilidade de grandes volumes de dados (Big Data) e o desenvolvimento de novos algoritmos de aprendizado de máquina (machine learning), especialmente as redes neurais profundas (deep learning), a IA passou por uma revolução significativa. Esse novo paradigma permitiu que sistemas computacionais aprendessem a partir de dados, melhorando seu desempenho ao longo do tempo sem programação explícita para cada tarefa (FERREIRA, 2024).

Atualmente, a IA está presente em diversas aplicações do cotidiano, como recomendações em plataformas de streaming, reconhecimento facial, assistentes virtuais e, especialmente, na automação de serviços, como os chatbots inteligentes. Essa evolução contínua da IA impulsiona a transformação digital em múltiplos setores, tornando-se um pilar estratégico para inovação, eficiência e competitividade nas organizações.

2.1.2 Aplicações no setor de serviços e atendimento

A aplicação da IA no setor de serviços e atendimento ao cliente tem promovido uma transformação significativa na forma como as empresas interagem com seus consumidores, oferecendo soluções mais ágeis, personalizadas e escaláveis. Com a crescente demanda por atendimento contínuo e eficiente, especialmente em canais digitais, a IA tem se destacado como uma ferramenta estratégica na automação de processos e no aprimoramento da experiência do usuário. Entre as principais aplicações estão os chatbots, assistentes virtuais e sistemas de recomendação, que utilizam técnicas de PLN e aprendizado de máquina para compreender solicitações, fornecer respostas contextuais e tomar decisões com base em dados históricos e comportamentais (BENIN, 2023).

No setor bancário, por exemplo, a IA é empregada para atendimento automatizado via chat, análise de risco de crédito e detecção de fraudes. Já no e-commerce, os sistemas inteligentes são utilizados para atendimento pré e pós-venda, rastreamento de pedidos, recomendações personalizadas e suporte técnico. Em empresas de telecomunicações, chatbots são amplamente adotados para resolução de problemas comuns, agendamento de serviços e renegociação de planos. Além disso, a integração de sistemas de IA com plataformas de CRM

permite um atendimento mais direcionado, adaptando as respostas com base no perfil e histórico do cliente. Essas aplicações não apenas reduzem custos operacionais, mas também aumentam a satisfação e fidelização do cliente ao proporcionar um serviço contínuo, eficiente e com menor tempo de espera (BENIN, 2023).

Assim, a IA se consolida como um elemento central na inovação dos serviços de atendimento, redefinindo o papel da tecnologia na construção de relacionamentos mais inteligentes e responsivos entre empresas e consumidores (BENIN, 2023).

2.2 Chatbots: Conceito, Tipos e Evolução histórica

Os chatbots são sistemas computacionais projetados para interagir com usuários por meio de linguagem natural, seja por texto ou voz. Eles se tornaram populares por sua capacidade de automatizar atendimentos, reduzir custos e melhorar a experiência do usuário (CASTRO DÍAZ, 2023). Nesta seção, serão apresentados os principais tipos de chatbots existentes, suas características, bem como uma breve retrospectiva de sua evolução tecnológica e funcional ao longo das últimas décadas.

2.2.1 Tipos de chatbots (Baseados em regras vs. Baseados em IA)

Esses sistemas podem ser classificados, de forma geral, em dois grandes tipos: chatbots baseados em regras e chatbots baseados em IA. Os chatbots baseados em regras funcionam a partir de fluxos conversacionais pré-definidos, nos quais cada interação do usuário leva a uma resposta programada, muitas vezes estruturada em árvores de decisão ou menus interativos diversos (CASTRO DÍAZ, 2023). Embora sejam eficientes para tarefas simples e repetitivas, como fornecer horários de funcionamento, informações sobre produtos ou status de pedidos, esses bots apresentam limitações em sua capacidade de compreender variações na linguagem ou contextos mais complexos, o que pode resultar em interações mecânicas e pouco flexíveis.

Por outro lado, os chatbots baseados em IA utilizam técnicas avançadas de PLN, aprendizado de máquina e, em muitos casos, redes neurais profundas

para interpretar e gerar respostas mais naturais e contextualmente relevantes. Esses sistemas são capazes de aprender com interações passadas, reconhecer intenções e entidades, adaptar-se ao estilo de comunicação do usuário e até mesmo lidar com ambiguidades na linguagem (CASTRO DÍAZ, 2023).

Um diferencial importante é a capacidade dos chatbots inteligentes de realizar inferências e oferecer respostas personalizadas com base em dados históricos e comportamentais do usuário. Com isso, eles se tornam ideais para aplicações mais complexas, como suporte técnico, triagem de problemas, orientações financeiras e até mesmo vendas consultivas. A distinção entre os dois tipos reflete não apenas o grau de sofisticação tecnológica, mas também os diferentes níveis de aplicação e valor agregado que podem oferecer ao atendimento ao cliente. A escolha entre um chatbot baseado em regras ou em IA depende, portanto, dos objetivos do negócio, da complexidade das demandas dos usuários e da experiência desejada na interação.

2.2.2 Evolução Histórica dos Chatbots

A evolução dos chatbots ao longo das últimas décadas reflete o progresso das tecnologias de IA e o crescente interesse em automatizar a comunicação entre humanos e máquinas. Os primeiros experimentos com agentes conversacionais datam da década de 1960, com o desenvolvimento do ELIZA, criado por Joseph Weizenbaum no MIT. ELIZA simulava um psicoterapeuta rogeriano utilizando simples regras de substituição de palavras e respostas padronizadas, sem qualquer compreensão real do contexto, mas demonstrou o potencial de engajamento humano com máquinas. Na década de 1970, surgiu o PARRY, um chatbot mais sofisticado que simulava um paciente com esquizofrenia paranoide, integrando aspectos rudimentares de comportamento emocional. Nos anos seguintes, entretanto, os avanços nessa área foram limitados, devido à falta de recursos computacionais e dados disponíveis, mantendo os chatbots como experimentos acadêmicos ou curiosidades tecnológicas (RIBEIRO; BONACIN; DOS REIS, 2024).

Foi apenas a partir da década de 2010, com o advento de smartphones, mensageiros instantâneos e, principalmente, com o avanço da IA e do PLN, que os chatbots passaram a ganhar protagonismo comercial. Plataformas como

Facebook Messenger, Telegram e WhatsApp começaram a permitir a integração de bots, enquanto empresas como Google, Amazon e Microsoft lançaram seus próprios assistentes virtuais, como Google Assistant, Alexa e Cortana. Paralelamente, surgiram frameworks especializados, como Dialogflow, Rasa e IBM Watson, que facilitaram o desenvolvimento de bots inteligentes, treináveis e integráveis com sistemas corporativos (RIBEIRO; BONACIN; DOS REIS, 2024).

A ascensão dos modelos de linguagem baseados em deep learning, como o BERT (Bidirectional Encoder Representations from Transformers) e, mais recentemente, os modelos da família GPT (Generative Pre-trained Transformer), revolucionou a forma como os chatbots compreendem e geram linguagem, oferecendo respostas mais naturais e adaptáveis. Atualmente, os chatbots representam uma tecnologia madura, amplamente utilizada em setores como e-commerce, saúde, educação e serviços financeiros, sendo fundamentais para estratégias de atendimento digital, automação e experiência do cliente. Essa trajetória evidencia como os chatbots evoluíram de simples scripts baseados em regras para sofisticados sistemas conversacionais, capazes de aprender, adaptar-se e oferecer suporte em interações cada vez mais complexas e humanas.

2.3 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural é uma área da IA que permite que máquinas compreendam, interpretem e gerem linguagem humana de forma significativa. Sua aplicação é essencial no desenvolvimento de chatbots mais inteligentes, capazes de compreender contextos e intenções dados (BENIN, 2023). Esta seção irá introduzir os fundamentos teóricos do PLN, além de apresentar os modelos e abordagens mais utilizados atualmente, como BERT e GPT.

2.3.1 Fundamentos de PLN

Os fundamentos do PLN envolvem uma combinação de linguística computacional, ciência da computação e estatística, permitindo que os sistemas

lidem com aspectos sintáticos, semânticos e pragmáticos da linguagem (BENIN, 2023).

Entre as tarefas básicas do PLN estão a tokenização (divisão do texto em unidades menores, como palavras ou frases), a lematização e stemming (redução de palavras à sua forma base), a análise morfossintática (identificação da função gramatical de cada termo), o reconhecimento de entidades nomeadas (como pessoas, organizações e locais), a análise de sentimentos e a classificação de intenções. Essas tarefas são fundamentais para que os sistemas compreendam o significado das entradas dos usuários, mesmo diante de ambiguidades, variações linguísticas ou ruídos na linguagem. Tradicionalmente, o PLN utilizava abordagens baseadas em regras e gramáticas formais, mas com o avanço do aprendizado de máquina, passaram-se a adotar modelos estatísticos e, posteriormente, redes neurais profundas que aprenderam padrões linguísticos a partir de grandes volumes de dados (BENIN, 2023).

Essa evolução permitiu uma significativa melhora na precisão e fluidez das interações homem-máquina. Atualmente, modelos como Word2Vec, GloVe, BERT e GPT revolucionaram o PLN ao introduzir representações vetoriais contextuais das palavras e capacidades de atenção, possibilitando que os sistemas não apenas reconheçam palavras isoladas, mas compreendam seu significado dentro de um contexto específico.

2.3.2 Modelos Populares

O desenvolvimento de aplicações baseadas em PLN depende de um conjunto de técnicas e algoritmos que permitem transformar dados linguísticos em informações estruturadas e compreensíveis para sistemas computacionais. Entre as técnicas fundamentais, destaca-se a tokenização, que consiste na divisão do texto em unidades menores, como palavras, frases ou subpalavras, servindo como ponto de partida para diversas outras tarefas. A análise sintática (ou parsing) é utilizada para identificar a estrutura gramatical das sentenças, revelando as relações entre os termos e auxiliando na compreensão do contexto e da intenção comunicativa. Já a classificação de intenção (intent classification) é uma etapa crucial no desenvolvimento de chatbots, pois permite identificar o

propósito da mensagem do usuário, como uma solicitação de informação, uma reclamação ou um pedido de ajuda. Essa tarefa é geralmente acompanhada pelo reconhecimento de entidades nomeadas (Named Entity Recognition – NER), que extrai elementos específicos do texto, como nomes, datas, locais ou produtos, essenciais para personalizar e direcionar a resposta (BENIN, 2023).

Além disso, algoritmos de análise de sentimentos e resolução de correferência contribuem para que o sistema compreenda nuances emocionais e referências indiretas dentro da conversa. Essas tarefas, que antes eram realizadas com base em regras e classificadores tradicionais, como Naive Bayes e SVM, hoje se beneficiam do uso de modelos de linguagem pré-treinados, que utilizam aprendizado profundo para extrair representações contextuais ricas da linguagem (BENIN, 2023).

Modelos como o BERT e o GPT têm se destacado nesse cenário. O BERT introduziu o conceito de atenção bidirecional, permitindo que o modelo entenda o significado de uma palavra com base em todo o contexto da sentença, e não apenas na sequência anterior. Já o GPT, com suas múltiplas versões (GPT-2, GPT-3 e GPT-4), trouxe avanços na geração de texto com coerência e fluidez, tornando-se referência para aplicações conversacionais complexas. Esses modelos são treinados com bilhões de parâmetros e grandes volumes de dados textuais, o que lhes confere uma capacidade sem precedentes de compreender e produzir linguagem natural, tornando-os essenciais na construção de chatbots inteligentes, capazes de oferecer interações naturais, contextualizadas e cada vez mais próximas do comportamento humano (RIBEIRO; BONACIN; DOS REIS, 2024).

2.4 Arquitetura de Sistemas de Chatbots Inteligentes

A construção de um chatbot inteligente envolve uma arquitetura composta por diversos elementos integrados, como interfaces de usuário, motores de IA, bancos de dados e APIs (Application Programming Interfaces). Compreender como essas partes se conectam e funcionam em conjunto é essencial para o desenvolvimento de soluções eficazes (RIBEIRO et al., 2024). Esta seção

aborda os componentes que formam essa arquitetura e a maneira como se comunicam com outros sistemas corporativos, como plataformas de CRM.

2.4.1 Componentes Principais

O primeiro componente é a interface de comunicação, responsável por viabilizar o contato direto entre o usuário e o chatbot. Essa interface pode estar integrada a canais como websites, aplicativos móveis, plataformas de mensagens (WhatsApp, Telegram, Facebook Messenger, entre outros) ou sistemas corporativos. Ela atua como ponto de entrada e saída das mensagens, transmitindo as entradas do usuário para os módulos internos do sistema e apresentando as respostas geradas (RIBEIRO et al., 2024).

O segundo componente essencial é o motor de IA (ou motor de conversação), que representa o núcleo inteligente do sistema. É nessa camada que ocorrem o reconhecimento de intenções, a extração de entidades e a geração de respostas, utilizando algoritmos de PLN, aprendizado de máquina e, muitas vezes, modelos de linguagem avançados como BERT ou GPT. O motor de IA é responsável por interpretar o que o usuário quer dizer e definir qual a melhor ação a ser tomada. Complementando esse processo está o banco de dados, que armazena informações relevantes para as interações, como históricos de conversas, perfis de usuários, dados de produtos ou serviços e regras específicas de negócio. Ele permite que o chatbot tenha memória, recupere informações contextuais e ofereça respostas personalizadas e coerentes com o cenário da conversa. Por fim, os integradores e APIs desempenham um papel fundamental na conexão do chatbot com sistemas externos, como CRMs, ERPs(Enterprise Resource Planning), gateways de pagamento, calendários, sistemas logísticos, entre outros (RIBEIRO et al., 2024).

Essa integração amplia significativamente as capacidades do chatbot, permitindo que ele realize ações transacionais, consultas em tempo real, atualizações de registros e acesso a serviços corporativos. A sinergia entre esses componentes é o que permite que um chatbot inteligente vá além de respostas automatizadas e ofereça uma experiência conversacional eficaz, fluida

e útil, atendendo às necessidades do usuário de forma personalizada e contextualizada.

2.4.2 Comunicação com APIs e Sistemas de CRM

A comunicação com APIs e sistemas de CRM é um dos aspectos mais estratégicos na arquitetura de chatbots inteligentes, pois é por meio dessa integração que o chatbot se conecta a dados corporativos e funcionalidades essenciais para oferecer um atendimento realmente personalizado e eficiente. As APIs funcionam como pontes que permitem ao chatbot acessar serviços externos ou internos, como sistemas de pagamento, bancos de dados de produtos, agendas, plataformas logísticas ou quaisquer outros serviços necessários para atender às demandas do usuário. Por exemplo, ao consultar o status de um pedido, o chatbot se comunica via API com o sistema logístico da empresa, recuperando as informações e apresentando-as ao cliente em tempo real (CHAGAS, 2024).

Já os sistemas de CRM, quando integrados ao chatbot, permitem que as interações sejam adaptadas com base no histórico de relacionamento com o cliente, seus dados de perfil, preferências e comportamentos anteriores. Isso possibilita desde saudações personalizadas até a sugestão de produtos com base em compras anteriores ou abertura de chamados com base em registros existentes. Além disso, essa integração torna possível registrar automaticamente novas interações no CRM, garantindo que todos os pontos de contato do cliente com a empresa fiquem documentados para futuras análises e decisões estratégicas (CHAGAS, 2024).

Para que essa comunicação ocorra de forma eficiente, é necessário garantir que os endpoints estejam bem estruturados, seguros e com baixa latência, garantindo uma troca de dados fluida durante a conversa. Tecnologias como RESTful APIs, Webhooks e OAuth são amplamente utilizadas nesse contexto para assegurar confiabilidade, escalabilidade e segurança na integração. Assim, a capacidade de se comunicar com APIs e sistemas de CRM transforma o chatbot de um simples canal de resposta em um agente ativo, capaz de executar tarefas, acessar informações críticas e agir como um

verdadeiro representante digital da empresa, reforçando a automação inteligente no atendimento ao cliente.

2.5 Tecnologias, Ferramentas e Frameworks

O desenvolvimento de chatbots inteligentes exige o uso de ferramentas adequadas que facilitem tanto a criação da lógica quanto o treinamento de modelos de linguagem. Atualmente, diversas plataformas e frameworks estão disponíveis para esse fim, além de bibliotecas especializadas que tornam o processo mais acessível e robusto (PACHECO, 2021). Nesta seção, serão apresentados os principais recursos utilizados na área, desde plataformas de desenvolvimento até linguagens de programação e bibliotecas específicas para IA e PLN.

2.5.1 Plataformas de Desenvolvimento

Entre as plataformas mais populares e amplamente utilizadas no mercado destacam-se o Dialogflow, do Google, que oferece uma interface intuitiva e recursos robustos de PLN, além de integração nativa com o Google Cloud e serviços como Google Assistant. O Dialogflow permite a criação de intents (intenções), entities (entidades) e fluxos conversacionais com facilidade, sendo ideal tanto para iniciantes quanto para projetos corporativos complexos (PACHECO, 2021).

Outra plataforma relevante é o Microsoft Bot Framework, que fornece uma infraestrutura completa para a construção, teste, publicação e gerenciamento de bots. Com suporte nativo ao Azure Bot Service, esse framework se destaca pela capacidade de integração com outros serviços da Microsoft, como o Dynamics 365 (CRM), Teams, e recursos de IA via Azure Cognitive Services, o que o torna uma solução altamente escalável e corporativa. Já o Rasa se diferencia por ser uma plataforma open source altamente personalizável, ideal para projetos que exigem maior controle sobre os dados, lógica de conversação e privacidade. O Rasa utiliza modelos próprios de aprendizado de máquina e permite a definição de políticas conversacionais com base em histórias de diálogo, oferecendo mais flexibilidade para implementações sofisticadas. Além dessas, outras ferramentas

como IBM Watson Assistant, Botpress, ManyChat e SnatchBot também são utilizadas em diferentes cenários, dependendo da complexidade do projeto, dos recursos necessários e da integração com sistemas legados (PACHECO, 2021).

A escolha da plataforma deve considerar fatores como suporte a múltiplos idiomas, disponibilidade de ferramentas de análise, capacidade de integração via API, suporte a modelos personalizados de IA e a curva de aprendizado para a equipe de desenvolvimento. Dessa forma, compreender as características e limitações de cada ferramenta é essencial para garantir que o chatbot atenda às necessidades específicas do negócio e ofereça uma experiência de usuário satisfatória e eficaz.

2.5.2 Linguagens e Bibliotecas

A construção de chatbots inteligentes exige o uso de linguagens de programação versáteis e bibliotecas especializadas em PLN e aprendizado de máquina. Nesse contexto, a linguagem Python destaca-se como uma das mais utilizadas devido à sua simplicidade, extensa documentação e ampla gama de bibliotecas voltadas para a IA. Python permite tanto a prototipação rápida quanto a construção de sistemas robustos em produção, sendo adotada em universidades, startups e grandes corporações (RESENDE, 2024).

Entre as bibliotecas mais relevantes está o TensorFlow, desenvolvido pelo Google, que oferece um ecossistema completo para o desenvolvimento e treinamento de modelos de aprendizado profundo, incluindo redes neurais recorrentes e transformadores, amplamente utilizados em aplicações de PLN. Complementando o TensorFlow, a biblioteca Keras facilita a criação de modelos com uma interface mais acessível, ideal para projetos que requerem flexibilidade e desempenho. Outra ferramenta importante é o spaCy, uma biblioteca de código aberto voltada para aplicações de PLN em produção, que oferece modelos pré-treinados eficientes para tarefas como tokenização, lematização, análise sintática, reconhecimento de entidades nomeadas e vetorização de palavras (RESENDE, 2024).

O spaCy é conhecido por sua velocidade e precisão, sendo uma excelente opção para aplicações em ambientes corporativos. Já a biblioteca NLTK (Natural Language Toolkit) é amplamente utilizada em ambientes acadêmicos e de

pesquisa, oferecendo uma vasta coleção de corpora linguísticos e ferramentas para análise textual, ideal para fins exploratórios e educativos. Além dessas, outras bibliotecas como Transformers da Hugging Face têm ganhado destaque por facilitar o uso de modelos pré-treinados de última geração, como BERT, RoBERTa e GPT, tornando-os acessíveis mesmo para desenvolvedores que não possuem amplo conhecimento em aprendizado profundo. A escolha da linguagem e das bibliotecas deve considerar a complexidade do projeto, os requisitos de desempenho, a necessidade de customização dos modelos e o ambiente de implantação (RESENDE, 2024). Com a combinação certa dessas ferramentas, é possível construir chatbots sofisticados, capazes de entender contextos, aprender com as interações e oferecer respostas cada vez mais humanas e assertivas.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta a metodologia adotada para o desenvolvimento do projeto, descrevendo o tipo de pesquisa conduzida, bem como o ambiente e os recursos utilizados para a construção do protótipo do chatbot inteligente. Além disso, são detalhados os materiais utilizados, como ferramentas de software e infraestrutura computacional empregada durante o desenvolvimento.

3.1 Tipo de Pesquisa

Esta pesquisa é de natureza bibliográfica e exploratória, contando também com uma etapa experimental para o desenvolvimento de um protótipo de chatbot. Conforme Gil (2017) explica, a pesquisa exploratória tem o objetivo de proporcionar uma compreensão mais profunda do tema, auxiliando no esclarecimento de questões e na abertura de caminhos para hipóteses que poderão ser investigadas em estudos futuros. O trabalho se propõe a examinar a evolução histórica e os avanços teóricos relacionados aos chatbots e à IA, demonstrando, por meio de um protótipo simples, como essas tecnologias podem ser aplicadas no atendimento ao cliente.

O primeiro passo será uma pesquisa bibliográfica, destinada a fornecer a base teórica necessária para o desenvolvimento do projeto. Essa revisão de literatura incluirá o estudo do desenvolvimento de chatbots, do PLN, e da evolução histórica da IA. De acordo com Severino (2018), a pesquisa bibliográfica envolve a análise de livros, artigos científicos e outras publicações relevantes, com o objetivo de criar uma base sólida para o desenvolvimento do trabalho.

A parte experimental do projeto consistirá na criação de um protótipo de chatbot simples. Após a coleta e análise das informações obtidas na revisão da literatura, será projetada a arquitetura do chatbot, que incluirá funcionalidades básicas capazes de responder a perguntas simples e simular interações com os usuários. Não haverá integração com sistemas reais, pois o foco será apenas demonstrar a viabilidade do uso de tecnologias de IA no atendimento automatizado.

Por fim, o processo de desenvolvimento será documentado de forma detalhada, incluindo a descrição das tecnologias utilizadas, os desafios enfrentados durante a implementação e sugestões para melhorias futuras.

3.2 Ambiente de Desenvolvimento e Recursos Utilizados

O desenvolvimento e os testes do protótipo foram realizados em um ambiente local, utilizando um computador pessoal. Para a implementação da aplicação, foi empregado o editor de código Visual Studio Code e a linguagem de programação Python, por meio de um ambiente virtual configurado com a ferramenta `venv`.

Entre as bibliotecas e frameworks utilizados no projeto, destacam-se:

- Flask: Responsável por estruturar a API que gerencia a comunicação do chatbot;
- spaCy: Utilizada para realizar o PLN;
- MySQL Connector: Para integrar a aplicação ao banco de dados MySQL;
- Twilio Sandbox: Ambiente de testes que simula a troca de mensagens via WhatsApp;
- Ngrok: Ferramenta utilizada para expor a aplicação local à internet, possibilitando que o webhook do Twilio se comunique com a API local de forma segura e funcional;
- MySQL Server: Utilizado como sistema de gerenciamento de banco de dados relacional para armazenar os registros das interações.

Durante os testes, a aplicação foi executada localmente e tornada acessível à internet por meio do Ngrok, uma ferramenta que cria um endereço público temporário. Esse endereço foi utilizado pela plataforma da Twilio para enviar as mensagens dos usuários ao chatbot em tempo real. Esse processo é viabilizado por um mecanismo chamado webhook, que consiste em uma "ponte de comunicação" automática entre dois sistemas — no caso, o WhatsApp (via Twilio) e o chatbot desenvolvido. O endpoint (ou ponto de acesso) é o endereço da aplicação onde essas mensagens são recebidas e processadas.

As interações foram realizadas utilizando o próprio número pessoal de WhatsApp, previamente vinculado ao ambiente de testes da Twilio. Essa

abordagem permitiu simular o funcionamento real do chatbot com usuários humanos, possibilitando a validação prática das funcionalidades implementadas.

4 RESULTADOS

Neste capítulo, são apresentados os principais resultados obtidos a partir da implementação do protótipo do chatbot, com base nas etapas previstas na metodologia. Descreve-se a sequência de desenvolvimento do sistema, abordando a definição das funcionalidades, a escolha das tecnologias empregadas e o processo de integração entre os diferentes componentes. O capítulo visa demonstrar como as decisões técnicas tomadas ao longo do projeto contribuíram para alcançar os objetivos propostos e validar, ainda que de forma experimental, o funcionamento do chatbot no contexto de atendimento automatizado.

4.1 Etapas do Desenvolvimento do Chatbot

O desenvolvimento do protótipo de chatbot seguiu uma sequência estruturada de etapas, visando garantir clareza na implementação e alinhamento com os objetivos definidos para o projeto. Inicialmente, foi realizado um levantamento de requisitos, mesmo em caráter simplificado, para identificar as funcionalidades mínimas que o chatbot deveria atender, como responder a saudações, consultar pedidos e fornecer informações básicas aos usuários. Embora o escopo seja experimental, essa etapa foi fundamental para delimitar as interações que seriam simuladas no ambiente de testes.

Com base nos requisitos levantados, foi feita a definição do escopo e das funcionalidades do protótipo, estabelecendo que o chatbot seria integrado ao WhatsApp através da API do Twilio, utilizando Python e o microframework Flask para o backend. As principais funcionalidades determinadas incluíram: receber e interpretar mensagens dos usuários, responder de forma automatizada a questões pré-definidas, registrar pedidos em um banco de dados MySQL e retornar mensagens de confirmação ou orientação.

Posteriormente, passou-se à modelagem da arquitetura do sistema, organizando a estrutura do código de forma modular, com a separação entre os componentes de processamento de mensagens, gerenciamento de rotas e utilidades auxiliares. Também foi planejado o uso do serviço Ngrok para expor o

servidor local à internet durante os testes, sem a necessidade de implantação em um servidor externo nesse primeiro momento.

Essa abordagem metodológica permitiu um desenvolvimento ágil, focado na demonstração prática do conceito de chatbot inteligente para atendimento ao cliente, em um ambiente realista e acessível.

4.2 Seleção de Tecnologias e Ferramentas

A seleção das tecnologias e ferramentas utilizadas no desenvolvimento do protótipo de chatbot teve como foco a viabilidade técnica, a facilidade de implementação e a adequação ao objetivo principal do projeto: demonstrar como a automação pode ser aplicada ao atendimento ao cliente por meio de um canal amplamente utilizado, o WhatsApp.

Com base nesses critérios, optou-se pela utilização do serviço de mensageria do Twilio, que permite a integração de aplicações com o WhatsApp através de APIs acessíveis e bem documentadas. Essa escolha garante que o protótipo opere em um ambiente real de comunicação, ampliando sua aplicabilidade e facilitando a simulação de interações com usuários em situações próximas às do uso comercial.

A linguagem de programação escolhida foi o Python, devido à sua simplicidade, vasta documentação e ampla adoção no desenvolvimento de chatbots e aplicações web. Para a construção da API de comunicação entre o WhatsApp e o sistema, utilizou-se o microframework Flask, que possibilita a criação de servidores web de maneira rápida e eficiente.

A estrutura do projeto foi organizada de forma modular, separando o código de acordo com suas responsabilidades específicas:

- Um módulo para o processamento e resposta das mensagens dos usuários.
- Um módulo para gerenciar rotas de comunicação entre o servidor e a API do Twilio.
- Um módulo de utilidades com funções auxiliares, como verificação de horário de funcionamento e mensagens padrão.

Para armazenamento dos pedidos realizados pelos usuários, foi integrado um banco de dados MySQL, permitindo que cada pedido fosse registrado de

forma segura e estruturada. O uso do MySQL reforça a robustez do protótipo e facilita futuras expansões do sistema, como relatórios de vendas ou integração com sistemas de gestão.

Além disso, o serviço Ngrok foi utilizado durante o desenvolvimento para expor o servidor local à internet, possibilitando testes reais no WhatsApp sem necessidade de uma infraestrutura de hospedagem inicial.

4.3 Modelagem da Arquitetura do Chatbot

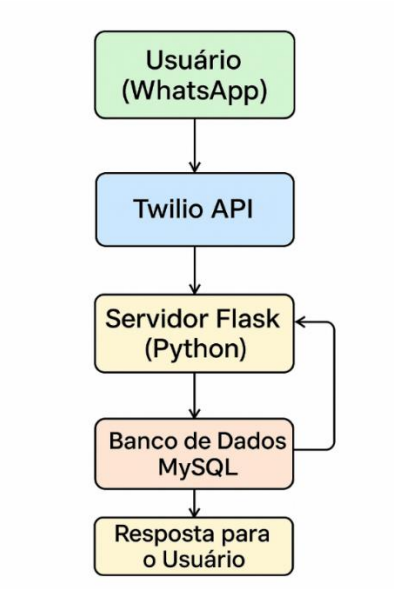
A arquitetura do protótipo de chatbot desenvolvido neste projeto foi estruturada de maneira modular e objetiva, com o propósito de construir um sistema funcional, de fácil compreensão e com potencial de expansão futura. O modelo proposto integra a comunicação via WhatsApp, o processamento de mensagens em um servidor backend e o armazenamento de dados em um banco de dados relacional, simulando, assim, um ambiente de atendimento automatizado.

Após o planejamento arquitetônico, o fluxo geral de funcionamento do sistema foi definido da seguinte forma: inicialmente, o usuário inicia uma interação enviando uma mensagem por meio do aplicativo WhatsApp. Em seguida, a mensagem é recebida pela API do Twilio, que a encaminha, via webhook HTTP, para o servidor backend, desenvolvido em Python utilizando o framework Flask.

O servidor backend é responsável por processar a mensagem recebida, realizando análises utilizando técnicas básicas de PLN, como a identificação de palavras-chave. Com base nessa análise, o sistema determina a intenção do usuário e gera uma resposta adequada, baseada em regras pré-definidas. Caso necessário, o backend realiza consultas ou armazenamentos de informações no banco de dados MySQL, como o registro de pedidos ou o histórico de interações. A resposta gerada é então encaminhada de volta à API do Twilio, que, por sua vez, a entrega ao usuário no ambiente do WhatsApp.

Essa organização modular possibilita uma comunicação fluida e independente entre os diversos componentes do sistema, além de facilitar futuras manutenções, escalabilidade e integrações com novos serviços. Na Figura 1, apresenta-se o diagrama geral da arquitetura do sistema:

Figura 1 - Diagrama geral do sistema



Fonte: Elaborado pelo autor.

O fluxo de interação entre o chatbot e o usuário foi concebido para ser simples, direto e funcional, atendendo aos propósitos de um protótipo de demonstração de conceito. A sequência típica de interação é organizada da seguinte forma:

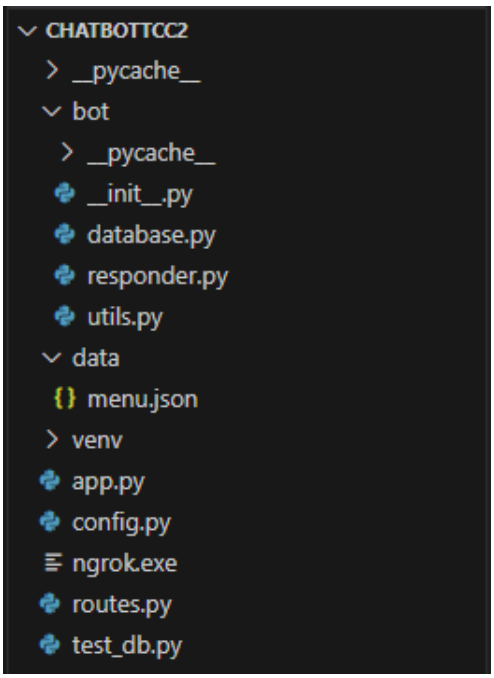
- O usuário envia uma mensagem inicial, como “Olá” ou “Gostaria de fazer um pedido”;
- O chatbot identifica a intenção expressa na mensagem e responde com uma saudação ou orientações específicas, como a solicitação de detalhes do pedido;
- Caso o usuário forneça informações adicionais, o sistema registra os dados no banco de dados e confirma a operação com mensagens de retorno, como “Seu pedido foi registrado com sucesso!”;
- Em situações nas quais a mensagem enviada pelo usuário não é reconhecida, o chatbot responde com mensagens padrão, solicitando uma reformulação ou oferecendo opções de ajuda.

Esse fluxo de interação busca oferecer uma experiência de comunicação eficiente e natural, reforçando a funcionalidade básica do chatbot no contexto de atendimento ao cliente.

5 ESTRUTURA DO PROTÓTIPO

A Figura 2 apresenta a estrutura geral do projeto de desenvolvimento do chatbot, organizada em diretórios e arquivos que separam claramente as diferentes responsabilidades do sistema. As principais pastas são `bot`, que contém os módulos responsáveis pela lógica do chatbot e pela interação com o banco de dados, `data`, que armazena arquivos de apoio como o menu em formato JSON, e `venv`, que contém o ambiente virtual do Python utilizado para a gestão das dependências do projeto. Além disso, arquivos principais como `app.py`, `routes.py` e `config.py` encontram-se na raiz do projeto, responsáveis pela inicialização do servidor, definição das rotas de comunicação e configuração do sistema, respectivamente.

Figura 2 - Estrutura completa do projeto



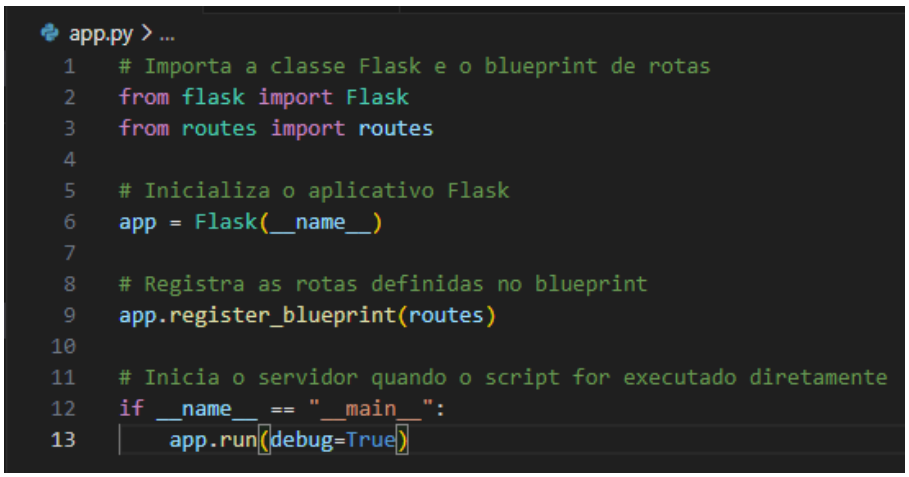
Fonte: do autor

Em seguida, serão descritos de forma individual os arquivos e módulos mais relevantes para o funcionamento do sistema, detalhando suas funções e suas relações dentro da arquitetura proposta.

`a) app.py`

O arquivo `app.py` é o ponto de entrada do projeto. Ele cria uma instância do servidor Flask, registra as rotas de comunicação do chatbot e inicia o servidor em modo de depuração para facilitar o desenvolvimento e testes, como mostrado na Figura 3.

Figura 3 - app.py

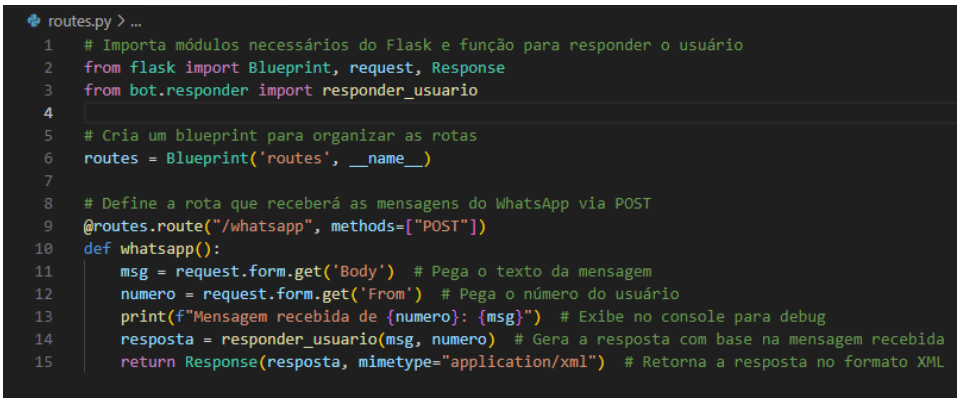


Fonte: do autor

b) `routes.py`

O arquivo routes.py define as rotas responsáveis pela comunicação entre o WhatsApp (via Twilio) e o chatbot. Quando o usuário envia uma mensagem, ela é capturada, processada e a resposta é retornada automaticamente em formato XML, conforme exigido pela plataforma de integração**,** como ilustrado na Figura 4.

Figura 4 - routes.py



Fonte: do autor

c) bot/responder.py

Na Figura 5 apresenta o trecho de código que realiza a importação das bibliotecas necessárias, incluindo o spaCy para PLN. Em seguida, carrega o modelo de linguagem em português, lê os itens do cardápio a partir de um arquivo JSON e inicializa o dicionário `CARRINHO`, que é responsável por armazenar os pedidos de forma temporária, vinculando cada usuário ao seu respectivo pedido.

Figura 5 - Cabeçalho, leitura do menu, definição do carrinho e carregamento do modelo PLN

Fonte: do autor

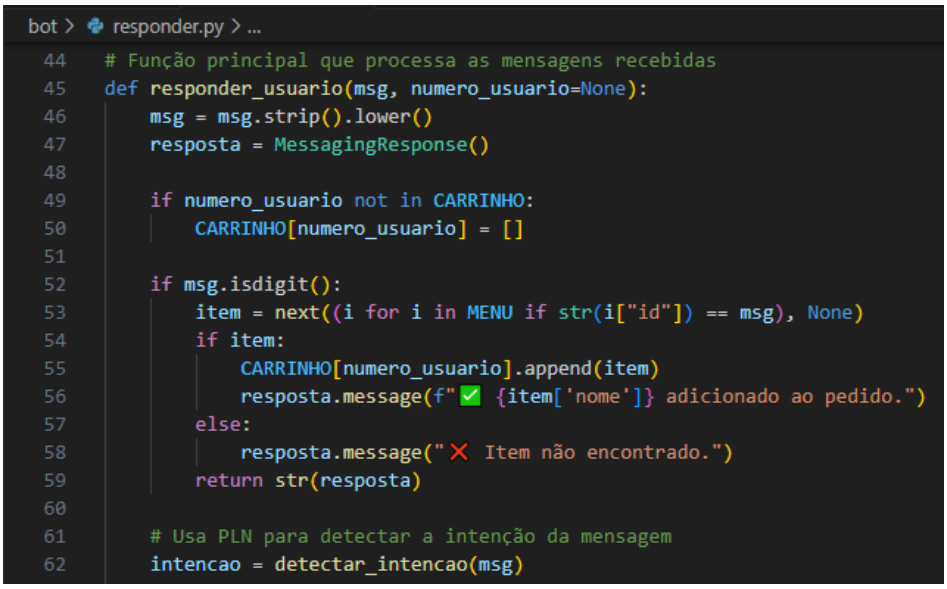
A função `montar_menu()` formata e retorna o cardápio completo da loja, contendo os nomes, preços e identificadores dos produtos. Já a função `detectar_intencao()` aplica o modelo de PLN para interpretar a mensagem do usuário, utilizando lematização para reconhecer palavras-chave e mapear a intenção da interação, como visualizar o menu, horário de funcionamento ou finalizar o pedido, conforme ilustrado na Figura 6.

Figura 6 - Funções montar_menu() e detectar_intencao()

Fonte: do autor

O início da função `responder_usuario()` normaliza a mensagem recebida, garante que o usuário esteja registrado no dicionário de pedidos e identifica se a entrada é numérica. Caso seja, o sistema interpreta como um pedido de item e o adiciona ao carrinho, retornando uma confirmação ao usuário, como mostrado na Figura 7.

Figura 7 - Função responder_usuario(): início do processamento e adição de itens ao carrinho



Fonte: do autor

Na figura 8, a função `responder_usuario()` continua com o processamento das mensagens, identificando a intenção e executando as ações correspondentes. Ela permite, por exemplo, consultar o menu, ver o resumo do pedido, confirmar a compra (com registro no banco de dados) ou retornar mensagens padrão quando a intenção não for reconhecida.

Figura 8 - Resumo do pedido, finalização e respostas padrão

```
64      if intencao == "ver_menu":
65          resposta.message(montar_menu())
66
67      elif intencao == "ver_horario":
68          status = "Estamos *abertos*!" if loja_aberta() else "Estamos *fechados* no momento."
69          resposta.message(f"{horario_funcionamento()}\n{status}")
70
71      elif intencao == "saudacao":
72          resposta.message(saudacao())
73
74      elif intencao == "ver_pedido":
75          if not CARRINHO[numero_usuario]:
76              resposta.message("🛒 Seu carrinho está vazio.")
77          else:
78              resumo = "📝 *Resumo do seu pedido:*\n"
79              total = 0
80              for item in CARRINHO[numero_usuario]:
81                  resumo += f"- {item['nome']} R${item['preco']:.2f}\n"
82                  total += item['preco']
83              resumo += f"\n💰 Total: R${total:.2f}"
84              resposta.message(resumo)
85
86      elif intencao == "finalizar_pedido":
87          if not CARRINHO[numero_usuario]:
88              resposta.message("Seu pedido está vazio. Envie o número do item para adicionar.")
89          else:
90              total = sum(item['preco'] for item in CARRINHO[numero_usuario])
91              itens = ", ".join(item['nome'] for item in CARRINHO[numero_usuario])
92              salvar_pedido(numero_usuario, itens, total)
93              resposta.message(f"✅ Pedido finalizado!\nItens: {itens}\nTotal: R${total:.2f}")
94              CARRINHO[numero_usuario] = []
95
96      elif intencao == "agradecimento":
97          resposta.message("😊 De nada! Estamos à disposição.")
98
99      else:
100         resposta.message("❓ Não entendi. Envie 'menu', 'horário', número do item ou 'pedido'.")
101
102     return str(resposta)
```

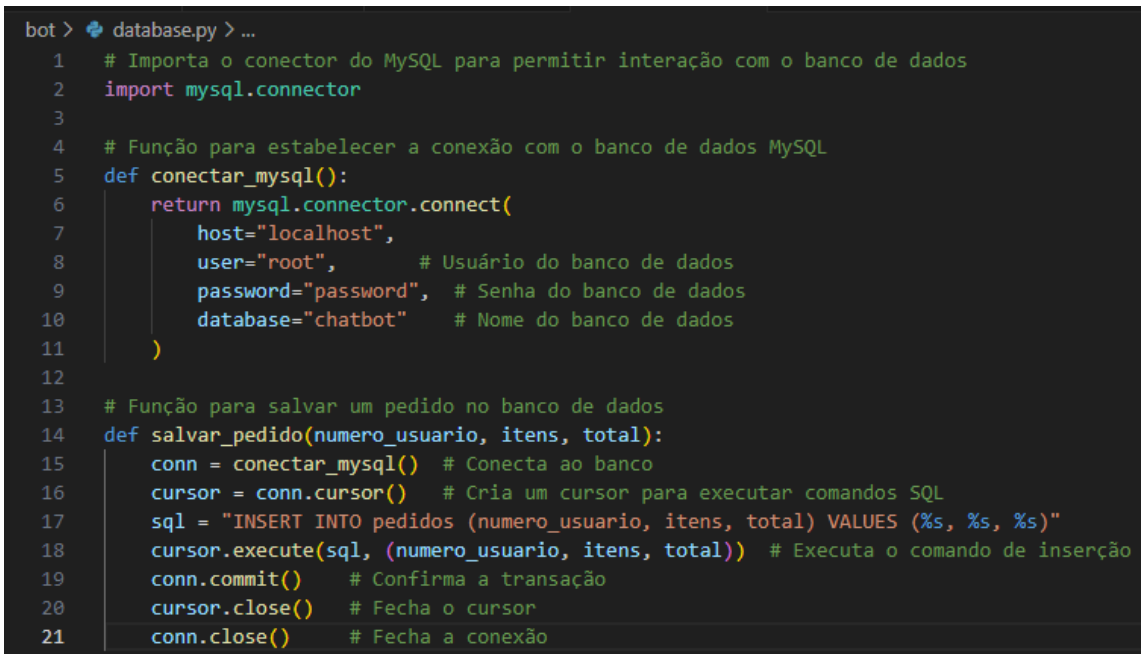
Fonte: do autor

d) `bot/database.py`

O módulo apresentado na Figura 9, é responsável por gerenciar a conexão entre o chatbot e o banco de dados MySQL. A função

`conectar_mysql` realiza a conexão com o banco especificado, enquanto `salvar_pedido` insere um novo pedido feito pelo usuário, registrando o número do telefone, os itens solicitados e o valor total no banco de dados.

Figura 9 - bot/database.py



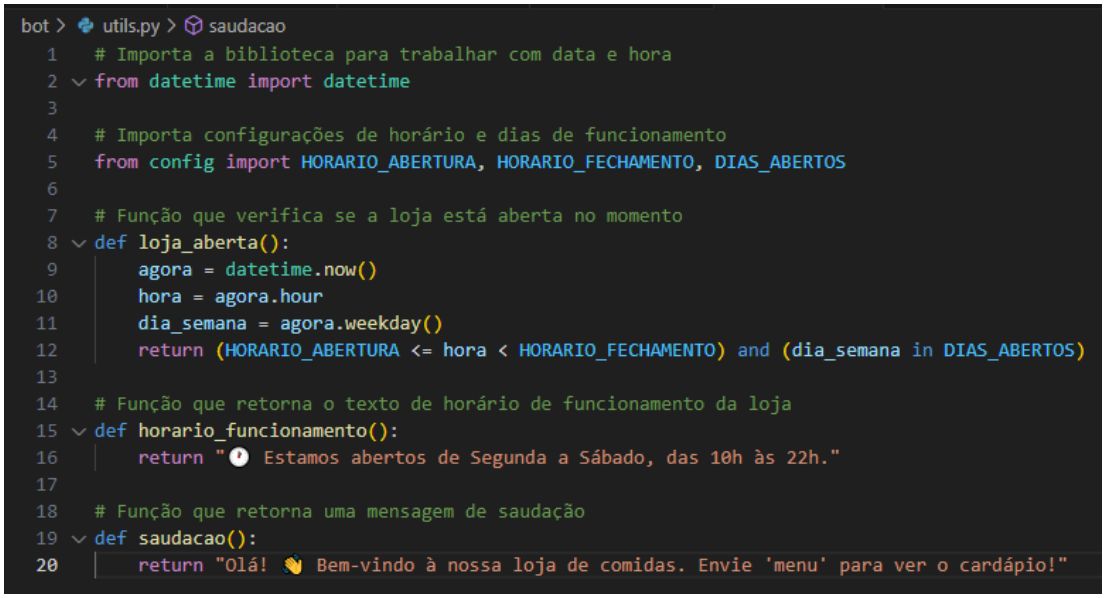
```
bot > database.py > ...
1   # Importa o conector do MySQL para permitir interação com o banco de dados
2   import mysql.connector
3
4   # Função para estabelecer a conexão com o banco de dados MySQL
5   def conectar_mysql():
6       return mysql.connector.connect(
7           host="localhost",
8           user="root",       # Usuário do banco de dados
9           password="password",  # Senha do banco de dados
10          database="chatbot"    # Nome do banco de dados
11      )
12
13  # Função para salvar um pedido no banco de dados
14  def salvar_pedido(numero_usuario, itens, total):
15      conn = conectar_mysql()  # Conecta ao banco
16      cursor = conn.cursor()   # Cria um cursor para executar comandos SQL
17      sql = "INSERT INTO pedidos (numero_usuario, itens, total) VALUES (%s, %s, %s)"
18      cursor.execute(sql, (numero_usuario, itens, total))  # Executa o comando de inserção
19      conn.commit()    # Confirma a transação
20      cursor.close()   # Fecha o cursor
21      conn.close()     # Fecha a conexão
```

Fonte: do autor

e) `bot/utils.py`

O módulo apresentado na Figura 10, agrupa funções auxiliares usadas pelo chatbot. A função `loja_aberta` determina se a loja está em horário de funcionamento baseado no dia e hora atuais. `horario_funcionamento` retorna o texto padrão informando o horário de atendimento, enquanto `saudacao` gera uma mensagem de boas-vindas para os usuários.

Figura 10 - bot/utils.py



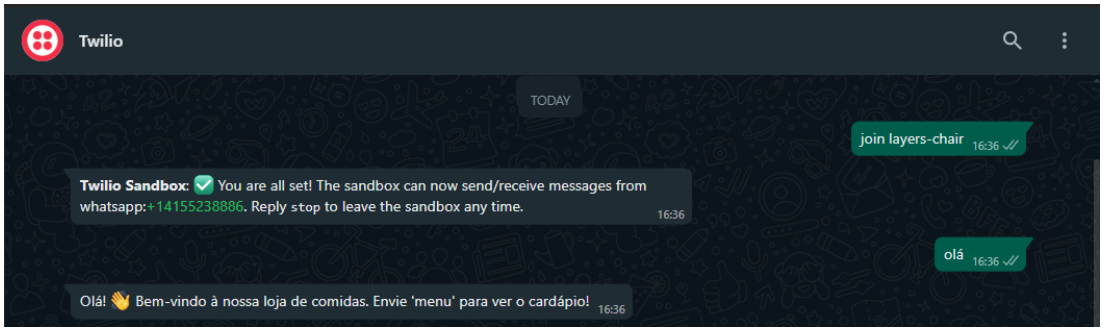
Fonte: do autor

5.1 Funcionamento do Protótipo

Nesta seção, são apresentados exemplos práticos da execução do protótipo de chatbot operando em ambiente real, via WhatsApp, com o uso da API do Twilio. As imagens a seguir ilustram tanto a interface de conversa entre o usuário e o bot quanto o registro automático das informações no banco de dados MySQL.

A mensagem “join layers-chair” é um código enviado automaticamente pelo sistema do Twilio ao vincular o número do WhatsApp ao ambiente de testes. Trata-se de um identificador aleatório utilizado para iniciar a sessão de testes, e não interfere no funcionamento do chatbot. Na sequência, o usuário inicia a conversa com uma saudação comum (“Olá”). O chatbot reconhece a intenção e responde com uma mensagem de boas-vindas e opções para continuar a interação, conforme ilustrado na Figura 11.

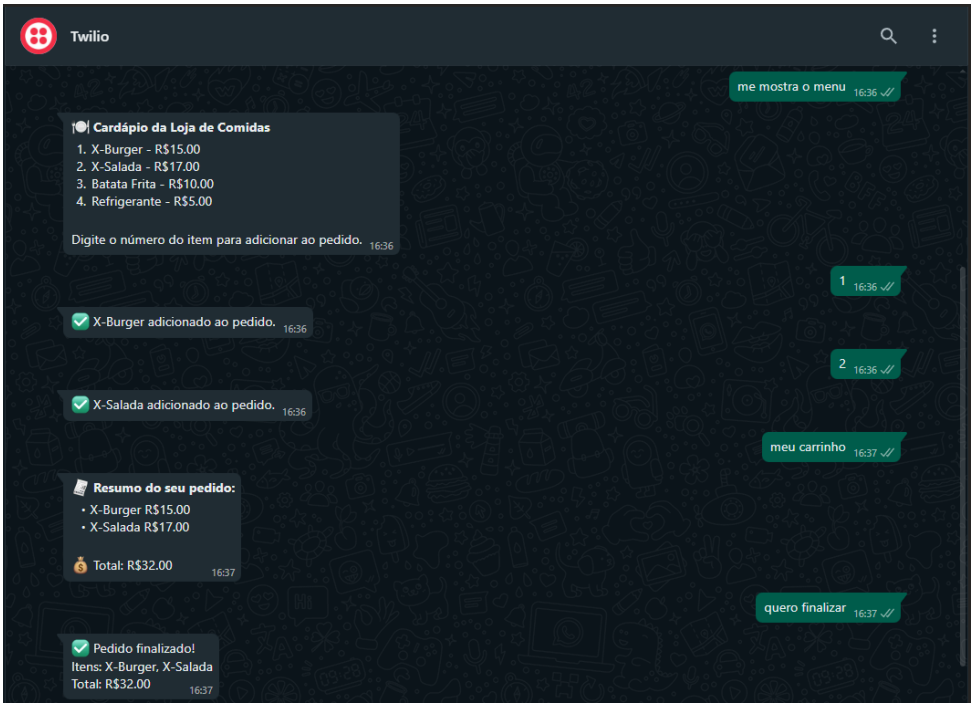
Figura 11 - Interação inicial com o chatbot via WhatsApp



Fonte: do autor

Nesta etapa, conforme ilustrado na Figura 12, o usuário demonstra interesse em realizar um pedido e o chatbot responde de forma dinâmica, interpretando a intenção da mensagem e fornecendo orientações necessárias para prosseguir. Ao consultar o cardápio, o usuário pode selecionar itens específicos por meio do número correspondente. O bot adiciona os itens ao carrinho temporário e fornece feedback confirmando a inclusão de cada produto. Após a seleção, o usuário pode finalizar o pedido com um comando apropriado, e o sistema registra automaticamente as informações no banco de dados, retornando um resumo com os itens solicitados e o valor total da transação.

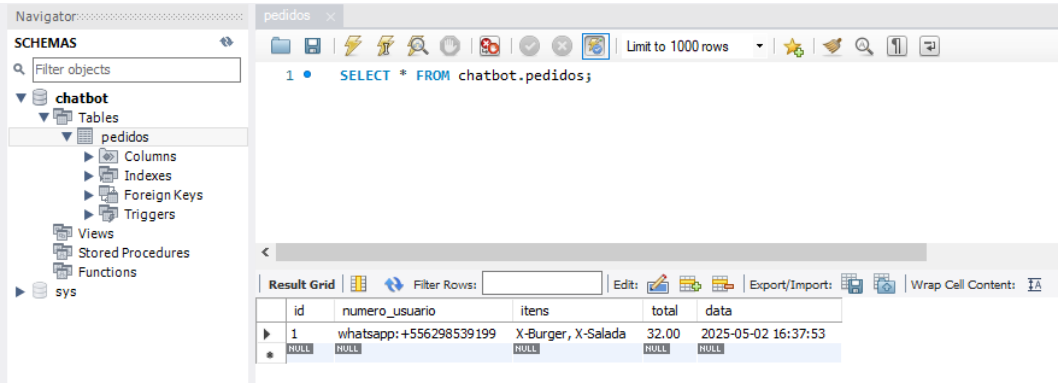
Figura 12 - Fluxo de solicitação e finalização de pedido pelo chatbot



Fonte: do autor

A Figura 13 apresenta a tabela de registros no banco de dados MySQL, contendo os pedidos realizados pelos usuários por meio do chatbot. As informações são armazenadas de forma estruturada após a conclusão de cada pedido, permitindo o acompanhamento das interações e garantindo a persistência dos dados relevantes.

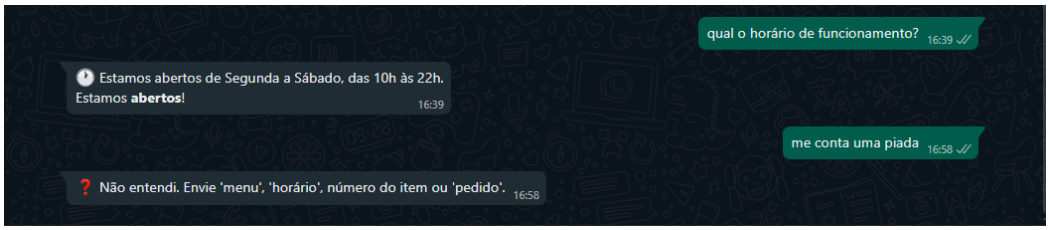
Figura 13 - Visualização dos registros no banco de dados MySQL



Fonte: do autor

Na Figura 14, são demonstrados exemplos de interações que não envolvem diretamente o processo de pedido, mas que fazem parte das funcionalidades adicionais do chatbot. Ao ser questionado sobre o horário de funcionamento, o bot reconhece corretamente a intenção e responde com a informação solicitada de forma clara e objetiva, reforçando se a loja está aberta ou fechada no momento. Em outro exemplo, o usuário envia uma mensagem fora do escopo esperado, solicitando uma piada. Como a intenção não é reconhecida pelas regras do sistema, o chatbot retorna uma mensagem padrão, orientando o usuário a utilizar comandos válidos como “menu”, “horário” ou “pedido”.

Figura 14 - Respostas do chatbot a comandos diversos e mensagens não reconhecidas



Fonte: do autor

5.2 Limitações do Protótipo

Apesar de o protótipo desenvolvido cumprir com sucesso seu objetivo principal — simular um atendimento automatizado via WhatsApp para pedidos de uma loja de comidas —, algumas limitações técnicas e funcionais ainda se fazem presentes e são relevantes para avaliação crítica e para direcionar possíveis evoluções futuras do projeto.

Do ponto de vista técnico, o modelo de PLN utilizado é baseado em regras simples com suporte da biblioteca SpaCy. Embora isso represente um avanço em relação a métodos puramente baseados em palavras-chave, o sistema ainda possui limitações na interpretação de frases mais complexas, ambíguas ou fora do contexto previamente mapeado. Isso pode levar a respostas genéricas ou incompreensão de solicitações atípicas, restringindo a naturalidade e a adaptabilidade do diálogo.

Em termos funcionais, o chatbot ainda carece de recursos como autenticação de usuários, integração com sistemas de pagamento, personalização de respostas com base em preferências anteriores e funcionalidades administrativas, como cancelamento de pedidos ou edição de itens no carrinho.

Durante o desenvolvimento, alguns desafios também foram enfrentados, principalmente relacionados à configuração do ambiente de testes com a API do Twilio, que exige um processo de verificação e vinculação entre o número do WhatsApp e o ambiente local através de ferramentas como o Ngrok. Também houve obstáculos na manipulação de dados com o banco MySQL, especialmente no que se refere à consistência e persistência dos dados entre sessões.

6 CONSIDERAÇÕES FINAIS

Este capítulo final apresenta uma reflexão geral sobre os resultados alcançados com o desenvolvimento do chatbot inteligente para atendimento ao cliente. São discutidas as limitações identificadas, as contribuições práticas do projeto e as perspectivas para aplicações futuras. Além disso, propõem-se sugestões de aprimoramento que podem ser consideradas em versões futuras da solução, visando aumentar sua eficiência e adaptabilidade em contextos reais de uso.

6.1 Avaliação Geral do Projeto

O desenvolvimento deste trabalho teve como principal objetivo a construção de um protótipo funcional de chatbot para atendimento automatizado via WhatsApp, utilizando a API do Twilio e técnicas de PLN. Ao longo do projeto, foram abordadas as etapas de concepção da arquitetura do sistema, implementação dos módulos fundamentais, integração com o banco de dados MySQL e validação do funcionamento por meio de testes práticos. O protótipo demonstrou ser eficaz na realização de interações simples com o usuário, como exibir o cardápio, registrar pedidos e apresentar o horário de funcionamento, oferecendo uma experiência fluida e direta.

A incorporação de PLN por meio do modelo de linguagem spaCy permitiu ao chatbot identificar intenções em mensagens textuais com maior flexibilidade, tornando a comunicação mais natural e próxima do comportamento humano. Isso representa um avanço significativo em relação a abordagens puramente baseadas em palavras-chave, ainda que limitado pela simplicidade do modelo e pela ausência de aprendizado contínuo.

Durante a execução do projeto, foram enfrentados desafios relacionados à configuração do ambiente de integração entre o Twilio e o Flask, bem como à manipulação de dados em tempo real com múltiplos usuários simultâneos. Tais obstáculos, no entanto, contribuíram para o amadurecimento técnico envolvido na construção de sistemas distribuídos e integrados com plataformas externas.

Embora o protótipo atenda aos objetivos definidos inicialmente, reconhece-se que há diversas possibilidades de aprimoramento, como a

implementação de um sistema de autenticação, maior robustez no entendimento de linguagem natural, uso de modelos de IA mais avançados e integração com meios de pagamento. Além disso, a escalabilidade do sistema pode ser aumentada com a migração para servidores em nuvem e uso de containers, como Docker.

Por fim, este trabalho contribui como base prática e teórica para estudantes e profissionais interessados na criação de chatbots voltados ao atendimento automatizado, demonstrando que, mesmo com recursos limitados, é possível construir soluções funcionais e aplicáveis ao contexto real de negócios.

6.2 Sugestões de Melhorias Futuras

Para tornar o chatbot mais robusto e adequado a cenários de produção, algumas melhorias podem ser consideradas em futuras versões. A principal delas seria a implementação de um modelo de PLN mais avançado, como o uso de redes neurais treinadas com exemplos reais de conversas, ou mesmo a integração com APIs externas especializadas em compreensão de linguagem natural, como o Dialogflow ou Rasa. Outra melhoria relevante seria a inclusão de autenticação por número de telefone, permitindo um controle mais refinado sobre usuários recorrentes, histórico de pedidos e personalização de mensagens.

Além disso, a integração com gateways de pagamento e serviços de logística agregaria valor comercial direto ao sistema, permitindo a realização de pedidos completos de ponta a ponta. Outras funcionalidades úteis seriam: permitir ao usuário editar ou cancelar pedidos, fornecer feedbacks sobre o atendimento, e disponibilizar métricas administrativas para o gestor da loja, como volume de pedidos por dia e itens mais solicitados.

Por fim, a criação de uma interface gráfica complementar para gerenciar o chatbot e acompanhar as interações em tempo real pode facilitar a administração do sistema por usuários sem conhecimento técnico, tornando a solução mais acessível e prática para diferentes tipos de negócios.

REFERÊNCIAS

BARBOSA, Lucia Martins; PORTES, Luiza Alves Ferreira. Inteligência artificial. *Revista Tecnologia Educacional*, Rio de Janeiro, n. 236, p. 16–27, jan./mar. 2023. Disponível em: https://abt-br.org.br/wp-content/uploads/2023/03/RTE_236.pdf#page=16. Acesso em: 2 mar. 2025.

BENIN, Keli Rodrigues do Amaral. *Processamento de linguagem natural e a Ciência da Informação: inter-relações e contribuições*. 2023. 103 f. Dissertação (Mestrado em Ciência da Informação) – Universidade Estadual de Londrina, Londrina, 2023. Disponível em: https://repositorio.utfpr.edu.br/jspui/bitstream/1/32098/3/processamentolinguagemnaturalcienciainformacao.pdf. Acesso em: 10 mar. 2025.

CASTRO DÍAZ, Cristian. Como os chatbots e as interfaces conversacionais estão mudando a forma como utilizamos tecnologia. *Seidor*, 24 maio 2023. Disponível em: https://www.seidor.com/pt-br/blog/como-os-chatbots-e-interfaces-conversacionais-estao-mudando-forma-como-utilizamos-tecnologia. Acesso em: 19 mar. 2025.

CHAGAS, Pedro Soethe. *Aplicação de chatbot em âmbito corporativo para a automação de processos em uma empresa de eletroeletrônicos industriais*. 2024. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Santa Catarina, Florianópolis, 2024. Disponível em: https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/256732/TCC.pdf?sequence=1&isAllowed=y. Acesso em: 10 abr. 2025.

CHATZIMPARMPAS, Angelos; FEDORYSHYN, Dmytro; CHO, Inria. Integration of AI in CRM: challenges and guidelines. *Journal of Innovation & Knowledge*, v. 8, 2023. Disponível em: https://www.sciencedirect.com/science/article/pii/S2199853123002536. Acesso em: 1 out. 2024.

INÁCIO, Carla Patrícia Almirante. *Os desafios da IA na comunicação das organizações*. 2022. Disponível em:

https://comum.rcaap.pt/bitstream/10400.26/51643/1/carla_in%C3%A1cio.pdf. Acesso em: 25 out. 2024.

MARKETS AND MARKETS. *Chatbot market by component, deployment, application, vertical and region – Global forecast to 2028.* 2023. Disponível em: https://www.marketsandmarkets.com/Market-Reports/chatbot-market-72302363.html. Acesso em: 27 set. 2024.

MELO, Nilmarcos Teodoro de; FREITAS, Michelle. CRM – gestão de relacionamento com o cliente: estudo de caso em uma cooperativa de crédito em Juína-MT. *Revista Científica da Ajes*, Juína, v. 8, n. 16, 2019. Disponível em: https://revista.ajes.edu.br/index.php/rca/article/download/218/174. Acesso em: 5 out. 2024.

MELLO FILHO, Luiz Lourenço de. *Assistentes virtuais como objetos de fronteira: um framework de modelagem pela Ciência da Informação*. 2023. Tese (Doutorado em Ciência da Informação) – Universidade de Brasília, Brasília, 2023. Disponível em: https://repositorio.unb.br/handle/10482/48825. Acesso em: 6 out. 2024.

OBJECTIVE. *Deep learning.* Curitiba: Objective, 2025. Disponível em: https://www.objective.com.br/insights/deep-learning/. Acesso em: 4 mar. 2025.

PACHECO, Fabricio Carvalho. *Estudo e desenvolvimento de um chatbot para automação de atendimento ao cliente.* 2021. 63 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Uberlândia, Uberlândia, 2021. Disponível em: https://repositorio.ufu.br/handle/123456789/36482. Acesso em: 13 abr. 2025.

RESENDE, Rafael Savignon de. *Criação de um chatbot para responder dúvidas sobre editais de concursos com processamento de linguagem natural.* 2024. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal do Espírito Santo, Vitória, 2024. Disponível em: https://repositorio.ifes.edu.br/handle/123456789/5610. Acesso em: 13 abr. 2025.

RIBEIRO, Cláudio Roberto; BONACIN, Rodrigo; DOS REIS, Julio Cesar. Uma abordagem para o uso de chats inteligentes e design instrucional como apoio à aprendizagem de programação. In: *SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE*, 35., 2024, Recife. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. Disponível em: https://sol.sbc.org.br/index.php/sbie/article/view/31387. Acesso em: 27 mar. 2025.

SILVA, João da; SOUZA, Maria de. *Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de LLM no TCMRio.* Natal: Universidade Federal do Rio Grande do Norte, 2024. Disponível em: https://repositorio.ufrn.br/items/2eaf113a-8403-4e3c-9ca1-972a89e0f0ed. Acesso em: 29 set. 2025.