UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS SOBRAL

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO

ÍTALO CARVALHO TEIXEIRA BARROS

GESTÃO DE CONTEXTO PARA INTELIGÊNCIA ARTIFICIAL GENERATIVA EM SOLUÇÕES EMPRESARIAIS

SOBRAL

25 de setembro de 2024

ÍTALO CARVALHO TEIXEIRA BARROS

GESTÃO DE CONTEXTO PARA INTELIGÊNCIA ARTIFICIAL GENERATIVA EM SOLUÇÕES EMPRESARIAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Iális Cavalcante de Paula Júnior

SOBRAL

25 de setembro de 2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

B284g Barros, Ítalo Carvalho Teixeira.
Gestão de contexto para inteligência artificial generativa em soluções empresariais / Ítalo Carvalho Teixeira Barros. – 2024.
58 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Engenharia da Computação, Sobral, 2024.
Orientação: Prof. Dr. Iális Cavalcante de Paula Júnior.

1. Inteligência Artificial Generativa. 2. Gestão de Contexto. 3. Suporte ao Cliente. 4. Embeddings em IA. I. Título.

CDD 621.39

ÍTALO CARVALHO TEIXEIRA BARROS

GESTÃO DE CONTEXTO PARA INTELIGÊNCIA ARTIFICIAL GENERATIVA EM SOLUÇÕES EMPRESARIAIS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Iális Cavalcante de Paula Júnior (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Jermana Lopes de Moraes
Universidade Federal do Ceará (UFC)

Prof. Me. Erick Aguiar Donato
Universidade Federal do Ceará (UFC)

À Deus, à minha família, aos meus amigos e professores.

AGRADECIMENTOS

À Deus por sempre estar comigo no meu caminhar, me dando suporte, forças, coragem e bençãos na minha vida.

Aos meus pais, Francisco Iran e Regina Claudia, que durante toda trajetória acreditaram em mim e no meu potencial, investindo na minha vida desde sempre, entregando qualidade de ensino, educação, e principalmente amor. Vocês são incríveis, e poder honrar vocês como meus pais é uma das dádivas que sempre serei grato.

Ao meu primo exemplo, Abraão Barros, que sempre me mostrou formas diferentes de enxergar a vida, noções de realidade que faziam eu acreditar que meus sonhos eram possíveis de serem alcançados, suporte na escuta durante momentos de decisões difíceis. Muito obrigado por seu cuidado e importância que você dar à minha trajetória profissional, espero honrá-lo com essa conclusão de curso e etapa de vida.

À minha avó Dededi (in memoriam) e à minha tia Helena que cuidaram de mim durante minha adolescência quando morei em Fortaleza para me dedicar aos meus estudos. Esse suporte foi essencial para a base do conhecimento estudantil que levei para a universidade.

Ao meu primo Matheus Barros por ser um exemplo de dedicação e disciplina, ferramentas necessárias para a construção de qualquer sonho de proporções grandes.

Ao Prof. Iális Cavalcante por me orientar não apenas nesse trabalho de conclusão, mas em vários momentos na minha trajetória acadêmica.

Ao meu amigo Julivan Hugo. Sua presença, conselhos e amizade enriqueceram minha jornada e contribuíram significativamente para este projeto.

À minha colega de formação Lucinara Fernandes, por todo nosso êxito nos nossos trabalhos ao decorrer de todo o curso.

“If you go out you come back
And you come back to an empty house
And now it’s loneliness again
It feels as though you’ve been dumped in the deep end
And there’s nobody there to rescue you
It’s just something that is thrown at you
You can’t throw it back to anybody
And all you can do is just carry on”

(Phaxe & Morten Granau - Lost)

RESUMO

Neste estudo, exploramos o impacto e as aplicações da gestão inteligente de contexto em Inteligência Artificial Generativa (IAG), especialmente na melhoria de sistemas de suporte ao cliente. Visto que, esta é conhecida por sua habilidade de gerar e adaptar conteúdo de forma autônoma e pode desempenhar um papel crucial no atendimento ao cliente. A gestão de contexto envolve identificar, interpretar e utilizar informações relevantes para assegurar que a Inteligência Artificial (IA) saiba responder de maneira precisa e contextualmente apropriada suas requisições com o menor custo operacional possível medido através dos *tokens*, que são unidades de processamento de IA. Uma gestão eficaz de *tokens* não apenas otimiza a resposta da IA, mas também controla custos operacionais, evitando o uso excessivo de recursos computacionais. Além disso, o trabalho aborda as funções de *embeddings* em IA, que são técnicas para transformar informações em vetores numéricos, facilitando assim a interpretação e classificação das informações. Este estudo demonstra como a IAG, através do uso gerencial de *tokens* e *embeddings*, pode contribuir com a gestão de contexto em ambientes de suporte ao cliente, oferecendo interações mais naturais e precisas com custo computacional reduzido.

Palavras-chave: Inteligência Artificial Generativa. Gestão de Contexto. Tokens em IA. Suporte ao Cliente. Embeddings em IA

ABSTRACT

This study explores the impact and applications of intelligent context management in Generative Artificial Intelligence (GAI), particularly in enhancing customer support systems. GAI is known for its ability to autonomously generate and adapt content, playing a crucial role in customer service. Context management involves identifying, interpreting, and utilizing relevant information to ensure that Artificial Intelligence (AI) can respond accurately and contextually to requests, while minimizing operational costs, measured through tokens, which are the processing units in AI. Effective token management not only optimizes AI responses but also controls operational costs by preventing the excessive use of computational resources. Additionally, the study addresses the role of embeddings in AI, techniques for transforming information into numerical vectors, thereby facilitating the interpretation and classification of information. This study demonstrates how GAI, through the strategic management of tokens and embeddings, can contribute to context management in customer support environments, offering more natural and precise interactions with reduced computational costs.

Keywords: Generative Artificial Intelligence. Context Management. AI Tokens. Customer Support. AI Embeddings

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
FAQs	Frequently Asked Questions
IA	Inteligência Artificial
IAG	Inteligência Artificial Generativa
PaaS	Platform as a Service
SGBDOR	Sistema de Gerenciamento de Banco de Dados Objeto-Relacional
SO	Sistema Operacional
TI	Tecnologia da Informação
UUID	Universal Unique Identifier

SUMÁRIO

1 INTRODUÇÃO

A emergência da IAG e sua aplicação em contextos empresariais, especialmente no atendimento ao cliente, representa uma transformação significativa nas interações comerciais contemporâneas (MAGALHAES; CASTRO, 2019). A ascensão dos chatbots, como evidenciado por estudos de Ana Regina Duarte (DUARTE, 2023) e da equipe da *COPPE UFRJ* (MAGALHAES; CASTRO, 2019), ilustra a crescente integração da IAG nas práticas de negócios. Estes sistemas de IAG, capazes de gerar e adaptar conteúdo de forma autônoma, têm se mostrado cruciais no suporte ao cliente, oferecendo respostas precisas e personalizadas em tempo real.

O advento dos *chatbots* e a sua progressiva sofisticação, conforme discutido por Duarte (DUARTE, 2023), introduzem novas dinâmicas na comunicação empresarial. A capacidade de fornecer interações mais humanizadas e contextualmente relevantes, com ganho de eficiência operacional, é um dos principais atributos dessas ferramentas. Estratégias de gestão de contexto é instrumental para identificar, interpretar e utilizar informações relevantes, assegurando respostas precisas, contextuais com menor custo computacional. Neste cenário, o uso eficiente de *tokens* e *embeddings* torna-se um aspecto central para otimizar a resposta da IA, controlando custos operacionais e aprimorando a experiência do usuário.

A pesquisa conduzida na *COPPE UFRJ* realça a importância de uma abordagem estratégica na adoção de *chatbots*, com uma ênfase na análise de como essas ferramentas são empregadas pelas empresas quando adotadas como solução no atendimento ao cliente. A integração efetiva dessas tecnologias exige não apenas um entendimento técnico, mas também uma apreciação das nuances de interações cliente-empresa e das expectativas dos consumidores.

Hohenstein (HOHENSTEIN *et al.*, 2023) conduziu uma análise que explora o impacto das respostas automáticas, um dos usos mais comuns da IA em comunicações digitais. O estudo evidencia que o emprego de respostas geradas por algoritmos (*smart replies*) modifica a maneira como indivíduos interagem e comunicam entre si, pois podem facilitar a comunicação, mas perder confiabilidade.

Ferramentas de IAG, como os sistemas de *ChatGPT*, desenvolvido pela *OpenAI*, estão sendo cada vez mais empregadas para gerar variados formatos de texto, desde e-mails e posts em redes sociais até códigos de software e redações. O levantamento sobre *smart replies* mostra que, embora o uso dessas tecnologias de IAG por parte das empresas possa ter benefícios, como acelerar a comunicação com o cliente e empregar termos emocionalmente carregados positivamente, também pode ser visto pelo destinatário (cliente) como insuficiente em termos de

transparência e engajamento pessoal.

Entretanto, é importante destacar que, com ajustes precisos e uma definição clara de objetivos, estas soluções podem transformar-se em ferramentas eficientes. Em particular, quando ajustadas para evitar atrasos na resposta ao usuário e para atendimentos mais simples de caráter informativo, essas respostas automatizadas podem se mostrar uma solução válida, mantendo a comunicação fluente e assegurando a satisfação do cliente.

1.1 Justificativa

Por conta da pandemia do Covid-19, houve um aumento considerável de tecnologias que evitam o contato humano e estimulavam o consumo de *e-commerces*. As vendas do *e-commerce* no Brasil, por exemplo, fecharam o ano de 2020 com alta de 73,88%, em relação a 2019, segundo os dados do índice MCC-ENET, desenvolvido pela *Câmara Brasileira da Economia Digital* (NASCIMENTO, 2021). Uma estimativa recente feita pela empresa de consultoria *Gartner* diz que, até final desse ano, 70% das interações com o cliente envolverão tecnologias de relação com a máquina, como os *chatbots*, isso equivale a um aumento de 15%, se comparado com 2018 (BEAT, 2021). Portanto, o atendimento automatizado nos canais digitais é uma prática importante para a comunicação entre empresas e clientes.

O exemplo mais marcante disso é a tecnologia de *chatbot* que, em 2020, teve seu mercado avaliado em US$ 17.17 bilhões, com previsão de chegar aos US$ 102.29 bilhões em 2026, segundo dados da *Mordor Intelligence* (BEAT, 2021). E assim, de acordo com Greg Bennett, diretor de design de conversação da *Salesforce*, o que mais se destaca no uso de *chatbots* é a sua capacidade de personalização. Um relatório da *Salesforce* diz que 95% dos líderes de Tecnologia da Informação (TI) têm priorizado essa tecnologia de automação, sendo que 70% deles observam uma economia de quatro horas por funcionário toda semana (BEAT, 2021).

As aplicações em *chatbots* no atendimento em todas as fases do funil de vendas em *e-commerces* estão cada vez mais presentes no relacionamento entre as marcas e seus respectivos clientes. Pesquisa da *Mobile Time* em parceria com a *Opinion Box* mostra que 78% dos consumidores entrevistados se comunicam com as empresas pelo *WhatsApp*, seguido por 51% pelo *Facebook Messenger* (LAM, 2022).

Esse cenário de *chatbots* foi influenciado pela abertura de uma Interface de Aplicações, ou *Application Programming Interface (API)*, em 2018, pelo *WhatsApp*, o que impactou no investimento em *chatbots* para melhorar a experiência do cliente em várias situações, indo

desde as informações sobre o produto até cancelamentos (LAM, 2022).

A plataforma *Facebook Developers* oferece uma API abrangente que permite a integração direta com o *WhatsApp Business*, possibilitando o uso de múltiplos atendentes, automação de atendimentos e obtenção de relatórios detalhados. Essa integração, oficializada pelo *Facebook*, assegura a confiabilidade e segurança na comunicação entre empresas e clientes, destacando a importância de soluções tecnológicas no atendimento ao consumidor (PLATFORMS, 2024).

Transformar a comunicação entre pessoas e marcas de forma simples e inteligente com tecnologia *chatbot*. Esse é o propósito da *Chatbot Maker*, *startup* cearense que recebe um investimento de R$1,5 milhão da *KPTL*. Os recursos são do Fundo *Criatec 3*, criado pelo *Banco Nacional de Desenvolvimento (BNDES)*. Entre os cotistas do Fundo, também estão: o *Banco do Nordeste (BNB)* e mais nove organizações (STARTUPI, 2021).

Contudo, em março de 2020 a empresa dá uma guinada com a criação da *Suri*, uma *chatbot* que catapulta a empresa de seus então 25 clientes para os cerca de 200 atuais – desde 2017 a empresa já criou mais de 450 *chatbots* diferentes. De lá para cá, foram mais de 1,5 bilhão de mensagens trocadas com a *Suri* por 3,5 milhões de pessoas (STARTUPI, 2021).

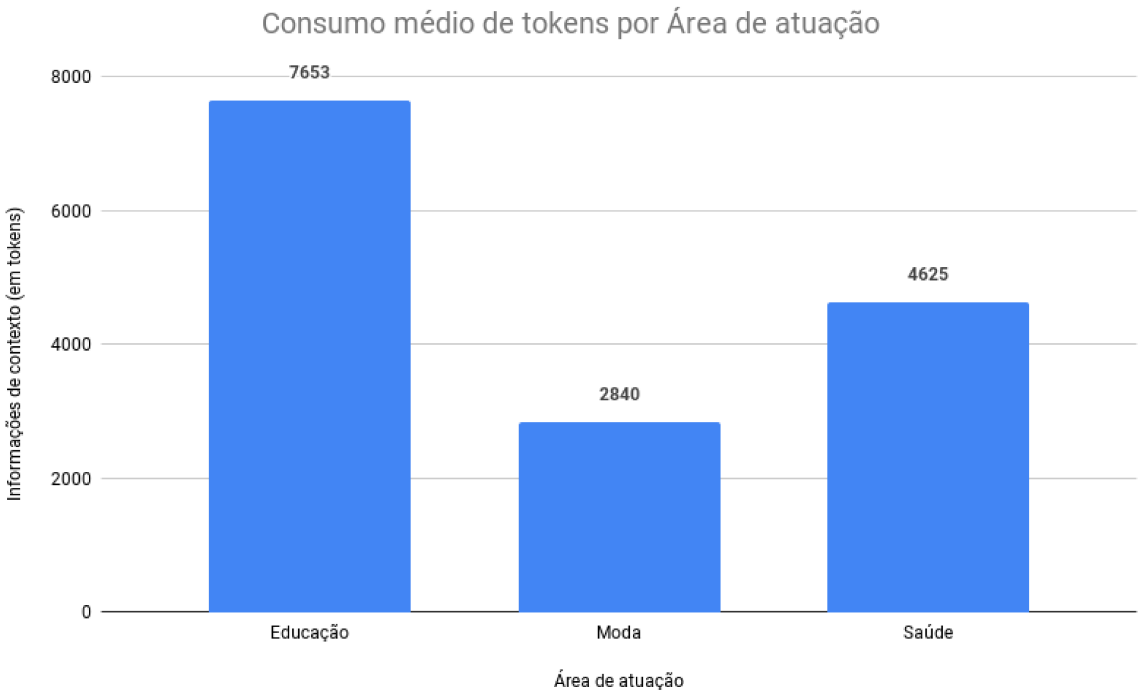
Fundada em 1998, *Take Blip* é uma empresa brasileira voltada para o atendimento ao cliente e a comunicação móvel. Com consolidação no mercado nacional e parcerias com grandes empresas globais, a *Take Blip* que desde 2016, quando entrou no mercado de soluções utilizando *chatbots*, foram desenvolvidos mais de 100 mil *chatbots* e mais de 10 bilhões de mensagens trafegadas pela plataforma (TAKEBLIP, 2020).

A *Iabots* surgiu como um projeto inovador construído durante o desenvolvimento desse trabalho, com o objetivo de criar uma solução personalizada de atendimento ao cliente utilizando a IAG. O trabalho focou na criação de bots específicos para empresas de diversas áreas de atuação, como Educação e Saúde, com o intuito de democratizar o uso de IAG no atendimento ao cliente. A abordagem personalizada adotada pela *Iabots* visa não só enriquecer a experiência do usuário, mas também otimizar os recursos das empresas, adaptando a tecnologia às necessidades específicas de cada cliente.

Durante o desenvolvimento deste projeto, foram coletados e analisados dados decorrentes de clientes reais da *Iabots* sobre o uso de *tokens* de processamento necessários para o funcionamento ideal dos *bots* em diferentes setores. Esses dados, representados na Figura 1 elaborados pelo autor com o projeto *Iabots*, oferecem uma visão detalhada sobre o consumo de recursos de IA e demonstram como uma gestão eficiente de contexto pode influenciar positivamente

na otimização dos custos operacionais.

Figura 1 – Consumo médio de *tokens* por Área de atuação em atendimentos ao cliente com IA Generativa



Fonte: Elaborado pelo autor

A análise da Figura 1 revela uma tendência interessante: a área de educação, com suas variadas consultas de metodologias e abordagens educativas, exige um volume maior de *tokens* para um atendimento efetivo, seguido pelas áreas de saúde e moda. Esta distribuição sugere que a complexidade das interações e a necessidade de uma resposta personalizada e contextualmente rica impactam diretamente no consumo de recursos de IA. Tais *insights* reforçam a importância de uma gestão de contexto eficiente, capaz de ajustar dinamicamente a alocação de *tokens* para maximizar a eficiência sem comprometer a qualidade do atendimento.

Portanto, a gestão de contexto, quando bem aplicada, permite não só a personalização do atendimento, mas também uma significativa economia de recursos, alinhando-se perfeitamente aos objetivos de sustentabilidade operacional e satisfação do cliente no ambiente empresarial moderno.

1.2 Trabalhos Relacionados

Nesta seção é explorado pesquisas e estudos que fornecem um contexto valioso e *insights* relacionados ao uso de tecnologias de Inteligência Artificial Generativa e *ChatGPT* no âmbito empresarial, focando em como essas inovações estão remodelando a comunicação e o atendimento ao cliente. Os trabalhos selecionados refletem não apenas o estado atual da tecnologia, mas também as oportunidades, desafios e riscos associados a sua implementação no ambiente de negócios.

1.2.1 Oportunidades, Riscos e Desafios do ChatGPT para Comunicação Empresarial (DUARTE, 2023)

Ana Regina da Silva Duarte investigou o impacto do *ChatGPT* na comunicação empresarial. Ela realizou uma revisão da literatura abrangente, incluindo várias fontes de periódicos científicos internacionais, o que ajuda a garantir a relevância de seu trabalho. A pesquisa de Duarte se concentrou na análise das implicações do *ChatGPT* no ambiente corporativo moderno, examinando tanto os riscos quanto as oportunidades que essa tecnologia apresenta.

O estudo se aprofunda nas potenciais estratégias que as empresas podem adotar para ultrapassar os desafios identificados, propondo caminhos para potencializar a utilidade do *ChatGPT* no longo prazo. A autora propôs uma reflexão sobre a necessidade urgente de abordagens inovadoras na integração do *ChatGPT* nas dinâmicas de comunicação corporativa, apoiando-se nos *insights* de George & George (2023), cujas ideias ilustram como a tecnologia pode fomentar interações mais personalizadas e dinâmicas, intensificando significativamente o engajamento dos clientes. (GEORGE *et al.*, 2023).

Além disso, Duarte sublinhou a importância da personalização nas interações digitais, uma tendência crescente no ambiente empresarial moderno. O estudo explora como a capacidade antropomórfica do *ChatGPT* pode redefinir a percepção dos usuários sobre as tecnologias de autoatendimento, tornando-as mais confiáveis e acessíveis. Ao destacar a habilidade do *ChatGPT* em fornecer recomendações personalizadas e inteligentes — baseadas em análises detalhadas de avaliações de sentimentos, preferências, opiniões e comportamentos dos consumidores —, a autora argumentou que esta tecnologia não somente eleva o padrão de serviços prestados, mas também impulsiona a produtividade e qualidade percebida pelo consumidor.

1.2.2 Avaliação do Uso de Chatbots pelas Empresas como Meio de Atendimento ao Consumidor (MAGALHAES; CASTRO, 2019)

Marcela Roméro Magalhães e Rayssa Bastos de Castro na *COPPE UFRJ*, fizeram uma análise detalhada sobre a aplicações empresariais voltadadas ao atendimento ao consumidor. O estudo visa entender como essas ferramentas digitais são implementadas nas estratégias de comunicação das empresas, identificando suas principais vantagens, tais como a eficiência na resposta e a disponibilidade 24/7, bem como suas limitações, incluindo a falta de compreensão de consultas complexas e a necessidade de treinamento contínuo para melhorar a precisão das respostas.

Com uma abordagem dualista, o estudo analisa as perspectivas financeira e do consumidor na utilização de *chatbots* pelas empresas. Financeiramente, evidencia-se que *chatbots* podem reduzir custos de atendimento ao cliente em até 30%, liberando atendentes de *call centers* de consultas repetitivas para focar em questões mais complexas, resultando em menos funcionários necessários e menor rotatividade. Em contrapartida, do ponto de vista do consumidor, ressalta-se a essencialidade de entender o perfil dos usuários para formar um grupo de teste representativo, assegurando que a ferramenta satisfaça as demandas antes de sua implementação real. O estudo também realça a importância de aferir a satisfação do cliente com os *chatbots*, sugerindo métodos para coletar e analisar *feedbacks* visando aperfeiçoamentos constantes. Discute-se sobre as tendências atuais e a crescente aceitação dos *chatbots*, sobretudo quando estes oferecem respostas que são rápidas, exatas e humanizadas.

Ambos os estudos proporcionam uma compreensão aprofundada sobre a integração de ferramentas de IA, como *chatbots*, nas práticas de comunicação empresarial e suporte ao cliente. Ao destacar o potencial de tecnologias avançadas, como o *ChatGPT*, integradas a plataformas de comunicação amplamente utilizadas, como o *WhatsApp*, esses estudos evidenciam a transformação na forma como empresas interagem com seus clientes. Essas pesquisas fornecem uma base importante para o desenvolvimento do presente trabalho, que visa não apenas explorar o uso da IA Generativa no contexto empresarial, mas também criar uma solução de chatbot competitiva e eficiente. Com base nos estudos de mercado e nas análises de aplicação do *GPT*, o trabalho atual estabelece um modelo que otimiza o uso de *tokens*, melhora a experiência do usuário e torna o sistema mais acessível e personalizável, atendendo às demandas empresariais de forma eficaz e econômica.

1.3 Objetivos

1.3.1 Objetivo Geral

Investigar e desenvolver uma Gestão de Contexto a fim de otimizar a eficiência e a eficácia de sistemas baseados em Inteligência Artificial Generativa, com foco especial em aplicações de suporte ao cliente.

1.3.2 Objetivos Específicos

- Analisar a evolução da IA Generativa: Examinar o desenvolvimento histórico e os avanços tecnológicos recentes na área de IAG, com ênfase em suas aplicações práticas;
- Estudar o processo de gestão de *tokens* em IA: Compreender como a gestão de *tokens* influencia a performance e a eficiência de sistemas de IA, identificando métodos para otimização de recursos;
- Explorar a implementação de IA Generativa no suporte ao cliente: Investigar como a IAG está sendo utilizada atualmente no suporte ao cliente. Incluindo estudo de casos de negócios que estão utilizando a solução;

2 FUNDAMENTAÇÃO TEÓRICA

A compreensão profunda da gestão de contexto em IA Generativa exige uma base sólida em diversas tecnologias subjacentes que possibilitam a implementação eficaz de soluções empresariais. Este capítulo visa desdobrar os conceitos-chave e as ferramentas tecnológicas essenciais que formam o alicerce para o desenvolvimento de sistemas avançados de atendimento ao cliente, focando na otimização de recursos e na personalização do atendimento.

2.1 *PostgreSQL*

O PostgreSQL é um Sistema de Gerenciamento de Banco de Dados Objeto-Relacional (SGBDOR) baseado no *POSTGRES*, Versão 4.2, desenvolvido no Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O *PostgreSQL* é um descendente de código aberto desse código original de Berkeley (POSTGRESQL, 2023). Na arquitetura do nosso projeto, utilizamos o *Postgres* para manipular diversas operações de dados fundamentais para a funcionalidade e eficiência do sistema.

Sua utilização neste projeto deve-se à capacidade de gerenciar eficientemente grandes volumes de dados e à sua extensibilidade. *PostgreSQL* permite a implementação de funcionalidades personalizadas que são cruciais para o tratamento de contextos dinâmicos em interações de IA, fazendo com precisão o seu papel nas respostas geradas ao cliente.

Dentro do contexto do nosso sistema, o *Postgres* é utilizado para armazenar e gerenciar dados de sessão do usuário, o que é crucial para uma experiência personalizada e contínua. Cada sessão é única e contém informações específicas do usuário, como histórico de interações, preferências e contexto atual de conversa. Isso permite que o sistema retome interações passadas sem perder o contexto, proporcionando uma experiência de usuário mais fluida e personalizada.

Dentro do contexto do nosso sistema, o *Postgres* é utilizado para armazenar os modelos de Frequently Asked Questions (FAQs), que são fundamentais para responder às dúvidas comuns dos usuários de forma eficiente. Ao utilizar *Postgres*, podemos estruturar e recuperar facilmente essas informações, permitindo que o sistema responda de forma consistente e precisa. Este aspecto será discutido em mais detalhes no próximo capítulo, onde abordaremos a estrutura e a aplicação dos modelos de FAQs dentro do nosso sistema.

Outro uso importante do *Postgres* no nosso projeto é armazenar e gerenciar a confi-

guração dos modelos de *GPT* (*GPTConfig*). Esta configuração determina como as respostas são geradas, personalizadas e otimizadas de acordo com os requisitos específicos do nosso sistema e as necessidades dos usuários. Ao armazenar essas configurações no *Postgres*, garantimos que elas possam ser atualizadas e consultadas de maneira eficiente, permitindo ajustes dinâmicos ao comportamento do modelo de IA com base nas interações do usuário e no *feedback* recebido de maneira personalizada para cada cliente. Este tópico também será explorado em maior profundidade no próximo capítulo, destacando a importância da configuração personalizada para melhorar a precisão e a relevância das respostas fornecidas pelo sistema.

2.2 *GoLang*

A linguagem *Go*, ou *GoLang*, é utilizada no *back-end* da aplicação e desempenha um papel crucial na manipulação de dados e na lógica de negócios. No contexto da nossa aplicação, *Go* é responsável pela implementação das operações *CRUD* (*Create, Read, Update, Delete*) que interagem com o banco de dados *PostgreSQL* para gerenciar modelos de *FAQs* e modelos *GPTConfig*. As vantagens do *Golang* são bem conhecidas por sua alta performance.

Os fatores importantes que contribuem para a alta performance do *Go* incluem a sua compilação direta para código de máquina, resultando em uma execução extremamente rápida. Além disso, *Go* possui um gerenciamento eficiente de memória com coleta de lixo (*garbage collection*) de baixa latência, concorrência simplificada através das *goroutines* e um modelo leve de *thread*, o que permite a execução simultânea de muitas tarefas com *overhead* mínimo. Essa combinação de características garante que *Go* possa lidar eficientemente com grandes volumes de dados e tráfego de rede intenso, tornando-o ideal para aplicações de alta demanda

2.3 *Heroku*

Trata-se de uma Plataforma como um Serviço, ou *Platform as a Service (PaaS)*, que facilita a hospedagem de aplicações na nuvem, oferecendo escalabilidade automática e gestão simplificada de infraestrutura. Isso permite aos desenvolvedores concentrarem-se na implantação das aplicações, pois facilita a configuração da infraestrutura para o *deploy* (IMAGINEDONE, 2023).

Heroku utiliza *dynos*, eles são basicamente contêineres virtuais do *Linux* usados para executar código com base em comandos do usuário. Os aplicativos podem ser escalados para

números específicos de *dynos* com base nos requisitos dos desenvolvedores (CLARK, 2023). Cada *dyno* opera de forma isolada, proporcionando um ambiente seguro e estável para a execução de diferentes componentes de uma aplicação.

Esses *dynos* podem ser escalados de forma independente, permitindo aos desenvolvedores aumentar ou diminuir recursos de acordo com a demanda, sem interromper o funcionamento do serviço. Essa característica é particularmente útil para lidar com picos de tráfego, garantindo que a aplicação permaneça disponível e responsiva. A plataforma também permite a configuração de *dynos* para tarefas específicas, como *web dynos* para atender solicitações *HTTP* e *worker dynos* para executar tarefas em *background*.

Heroku usa Buildpacks para definir como as imagens devem ser configuradas e slugs, que são cópias comprimidas de sua aplicação associadas a um dyno para tempo de execução.

2.4 *Docker*

O *Docker* é uma plataforma de software livre que permite aos desenvolvedores desenvolver, implementar, executar, atualizar e gerenciar componentes de contêineres executáveis e padronizados que combinam o código-fonte de aplicativos com as bibliotecas e estruturas do Sistema Operacional (SO) necessárias para executar o código em qualquer ambiente (IBM, 2023).

Utilizando a infraestrutura do *Heroku*, a aplicação pode ser escalada facilmente, beneficiando-se da alta disponibilidade, gerenciamento de *logs*, monitoramento e outras características essenciais para aplicações em produção. *Docker* e *Heroku* permitem encapsular aplicações em unidades autocontidas e independentemente implantáveis, fundamentais para um design baseado em Microserviços. *Docker* é popular pela sua portabilidade e abertura, enquanto Heroku é preferido por sua escalabilidade, segurança e desenvolvimento rápido. (FAKHRUDDIN, 2016)

Uma característica altamente eficiente do *Docker* são os *Dockerfiles*. Esses arquivos automatizam o processo de criação de imagens no *Docker*, servindo como um roteiro que define cada etapa necessária para montar uma imagem. O *Dockerfile* consiste em uma lista de instruções padronizadas, executadas pelo *Docker Engine*, que garantem a consistência do processo de construção da imagem, independentemente do conteúdo ou das variáveis de ambiente. Dessa forma, as imagens resultantes tornam-se unidades autocontidas, encapsulando tudo o que é necessário para rodar uma aplicação, enquanto os *containers* representam as instâncias em

execução dessas imagens, assegurando portabilidade e eficiência na execução das aplicações em qualquer ambiente.

A utilização do *Heroku*, em conjunto com o *Docker*, amplifica a eficiência e a flexibilidade do projeto, especialmente no gerenciamento dos *dynos*. O *Docker* permite encapsular a aplicação em contêineres, tornando a distribuição e escalabilidade mais simples e eficaz. Esse método de *containerização* complementa perfeitamente os *dynos* do *Heroku*, otimizando a implantação de atualizações e a gestão de recursos, garantindo um desempenho consistente e confiável da aplicação na nuvem (FAKHRUDDIN, 2016)

2.5 *Arquitetura Cliente-Servidor e Multilocação (Multitenancy)*

A arquitetura cliente-servidor é um modelo de computação distribuída que segmenta as funções em dois programas distintos: o cliente e o servidor. O cliente é responsável por enviar pedidos ao servidor, que por sua vez processa esses pedidos e retorna os resultados. Esta arquitetura é fundamental na construção de aplicações web e móveis, pois permite uma comunicação eficiente e escalável entre o *front-end* (cliente) e o *back-end* (servidor) (AWARI, 2023).

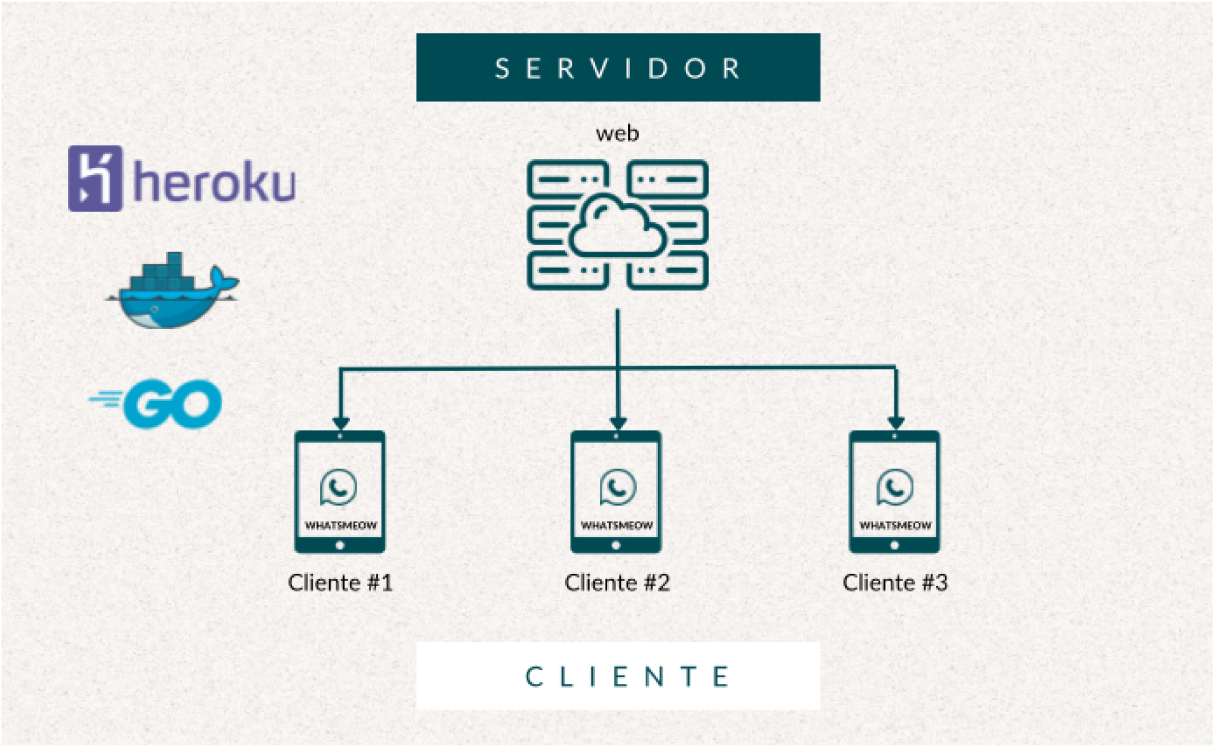
Na arquitetura cliente-servidor, o servidor hospeda os recursos e serviços, como bases de dados, arquivos, e aplicações de negócios, que são acessados pelos clientes. Esta estrutura oferece uma série de vantagens, incluindo a centralização do controle de acesso, a redução de redundâncias de dados, e a facilidade de manutenção e atualização de serviços. Além disso, promove uma divisão clara entre a interface do usuário e a lógica de processamento, facilitando o desenvolvimento e melhorando a experiência do usuário.

Multilocação, ou *multitenancy*, é um princípio de arquitetura de software que permite a uma única instância de uma aplicação servir múltiplos clientes, ou "inquilinos". Cada inquilino opera de forma isolada, de modo que suas configurações, dados e personalizações são mantidos separados uns dos outros. Esta abordagem é especialmente útil em ambientes de nuvem, onde a escalabilidade e a eficiência de recursos são cruciais (PEREIRA, 2019).

A implementação de *multitenancy* em uma arquitetura cliente-servidor oferece benefícios significativos, como a redução de custos operacionais e a eficiência no uso de recursos, já que várias instâncias do cliente compartilham a mesma infraestrutura e serviços do servidor. Além disso, , pois as mudanças feitas na aplicação ou no servidor afetam imediatamente todos os inquilinos. Contudo, requer cuidados especiais em termos de segurança de dados e isolamento

de inquilinos para evitar vazamentos de informações entre eles.

Figura 2 – Arquitetura cliente-servidor com *multitenancy*



Fonte: Elaborado pelo autor

A Figura 2 é uma representação gráfica da arquitetura cliente-servidor com *multitenancy*, onde cada cliente possui seu identificador único que garante o comportamento adequado nas respostas ao se comunicar com o servidor. Essa arquitetura desempenha um papel crucial na gestão eficiente de múltiplos clientes em um sistema distribuído, permitindo a escalabilidade e a personalização dos serviços de forma segura e eficaz.

2.6 *OpenAI API*

A *OpenAI API* é uma interface poderosa para interagir com modelos avançados de inteligência artificial desenvolvidos pela *OpenAI*, incluindo variações do famoso modelo *GPT (Generative Pretrained Transformer)*. Esses modelos podem ser configurados de diferentes maneiras para atender a diversos requisitos e contextos, tornando-os versáteis para uma ampla gama de aplicações (ALVES, 2023).

Uma das características mais notáveis da *OpenAI API* é a sua capacidade de usar diferentes versões do modelo *GPT*. Cada versão tem suas próprias características, como tamanho,

capacidade de entendimento, geração de texto, e eficiência em termos de custo e tempo de resposta. Essas variações permitem que desenvolvedores escolham o modelo mais adequado para suas necessidades específicas, equilibrando entre qualidade de resposta e recursos disponíveis.

A configuração do modelo é outra faceta importante da *OpenAI API*, permitindo aos usuários ajustar parâmetros como a temperatura (que controla a aleatoriedade das respostas), *max_tokens* (o comprimento máximo da resposta) e *top_p* (que ajuda a filtrar as possibilidades de resposta baseadas em sua probabilidade). Essas configurações são fundamentais para personalizar as respostas do modelo para se adequarem ao contexto e aos objetivos específicos do usuário (ALVES, 2023).

Um componente crucial na utilização da *OpenAI API*, especialmente para aplicações que possuem modelos do tipo *FAQ*, são os *embeddings*. *Embeddings* são representações vetoriais de uma entrada parametrizada que pode ser facilmente consumida por modelos e algoritmos de aprendizado de máquina (OPENAI, 2023). No contexto de *FAQs*, os *embeddings* são utilizados para compreender e mapear perguntas às suas respostas mais relevantes. Esse processo envolve a transformação de perguntas e respostas em vetores, permitindo identificar correspondências de alta similaridade entre eles. O funcionamento técnico dessa abordagem será detalhado mais profundamente no Capítulo 3.2.1, onde exploraremos a metodologia utilizada para o mapeamento vetorial, destacando as técnicas de similaridade e o uso dos *embeddings* na estruturação eficiente de respostas.

Em resumo, a *OpenAI API* fornece uma poderosa ferramenta para incorporar capacidades de inteligência artificial avançada em aplicações. Seus diferentes modelos, configurações ajustáveis e o uso de *embeddings* tornam-na particularmente eficaz para construir sistemas de resposta a perguntas, como serviços de *FAQ* automatizados, onde precisão e relevância são fundamentais.

2.7 *WhatsApp Business API*

WhatsApp Business API oferece uma plataforma robusta para empresas que buscam aprimorar a comunicação com clientes em grande escala. Esta API especializada facilita a automatização de interações, organiza fluxos de mensagens e viabiliza respostas rápidas, elementos fundamentais para sustentar expectativas modernas de atendimento instantâneo.

No contexto deste projeto, a *WhatsApp Business API* é explorada de maneira estudantil e exploratória, visando entender como a automação e personalização do atendimento

ao cliente podem ser ampliadas em ambientes empresariais de grande porte. A capacidade de gerenciar grandes volumes de mensagens simultâneas torna esta ferramenta particularmente adequada para organizações com alto tráfego de comunicação, contrastando com soluções mais adaptadas a empresas de menor dimensão.

Além disso, a integração da *WhatsApp Business API* proporciona uma oportunidade para analisar a eficácia das respostas automatizadas e do suporte personalizado, sob a égide do suporte fornecido pela *Meta*. Este enfoque permite uma investigação detalhada sobre a adaptabilidade da API a diferentes tamanhos e perfis de empresa, examinando como as estratégias de comunicação podem ser escaladas e personalizadas para atender a uma variedade de necessidades empresariais (PLATFORMS, 2024).

2.8 *Whatsmeow (Go package)*

Whatsmeow é um pacote *Go* específico para a interação automatizada com o *WhatsApp*, especialmente projetado para facilitar a comunicação entre empresas e seus clientes. Oferecendo recursos como envio e recepção de mensagens, gestão de grupos e notificações, torna-se uma ferramenta valiosa para empresas de pequeno e médio porte. A escolha desse pacote reflete a intenção de aplicar análises de estudo de caso focadas em melhorar a interação cliente-empresa através de canais digitais populares.

Este pacote funciona de maneira análoga ao *WhatsApp Web*, permitindo que os usuários gerenciem suas contas *WhatsApp* de forma eficiente através de instâncias autenticadas e seguras. O *Whatsmeow* mantém uma sincronização constante com as contas dos usuários, assegurando a atualização em tempo real de mensagens e dados.

Integrar *Whatsmeow* com *Heroku* e seus *dynos* proporciona uma arquitetura robusta de cliente-servidor e suporta eficientemente o conceito de *multitenancy*. Cada instância única do *WhatsApp* operada pelo *Whatsmeow* pode ser associada a um *dyno* individual no *Heroku*, permitindo escalabilidade e isolamento entre as sessões de clientes diferentes. Essa abordagem melhora significativamente a capacidade de resposta e a personalização do serviço ao cliente, enquadrando-se perfeitamente nas práticas modernas de atendimento digital e na infraestrutura de *cloud*.

3 METODOLOGIA

Este capítulo descreve a metodologia adotada para o desenvolvimento do projeto, que foi estruturado em quatro fases principais: Conceituação, Planejamento, Implementação e Gerenciamento. Estas etapas foram cuidadosamente delineadas para cobrir desde a formulação inicial da ideia até o gerenciamento e avaliação dos resultados. A Figura 3 apresenta a divisão temporal dessas etapas ao longo do período de desenvolvimento do projeto, entre 2023 e 2024, fornecendo uma visão clara da progressão e das prioridades em cada fase.

Figura 3 – Metodologia

Metodologia				2023				2024	
Fases	Objetivo	Foco	Descrição	1º TRI	2º TRI	3º TRI	4º TRI	1º TRI	2º TRI
Conceituação	Definir o escopo e os objetivos gerais do projeto	Escopo e objetivos	Esta fase envolve o entendimento inicial do problema, a definição dos objetivos do projeto e a concepção da solução geral relacionada com a evolução da perfomance das IAGs.	X					
Planejamento	Elaborar um plano detalhado das atividades, recursos necessários e cronograma	Planejamento e recursos	Nesta etapa, são planejadas todas as atividades necessárias, incluindo a seleção e estudo das tecnologias utilizadas, revisão bibliográfica e pesquisa de mercado.		X	X			
Implementação	Executar o plano, desenvolvendo as soluções propostas e integrando as tecnologias	Execução e desenvolvimento	Durante a implementação, as tecnologias são integradas e as soluções são desenvolvidas e testadas para garantir que atendam às necessidades identificadas. O objetivo dessa fase é trazer um Produto Viável Mínimo (MVP) suficiente para garantir novos usuários com uma boa experiência de consumo.			X	X		
Gerenciamento	Monitorar e controlar o projeto para garantir a qualidade e o alinhamento com os objetivos	Monitoramento e controle	O gerenciamento envolve a supervisão do projeto para garantir que esteja progredindo conforme planejado, fazendo ajustes conforme as necessidades dos clientes e implementação de novas funcionalidades.					X	X

Fonte: Elaborado pelo autor

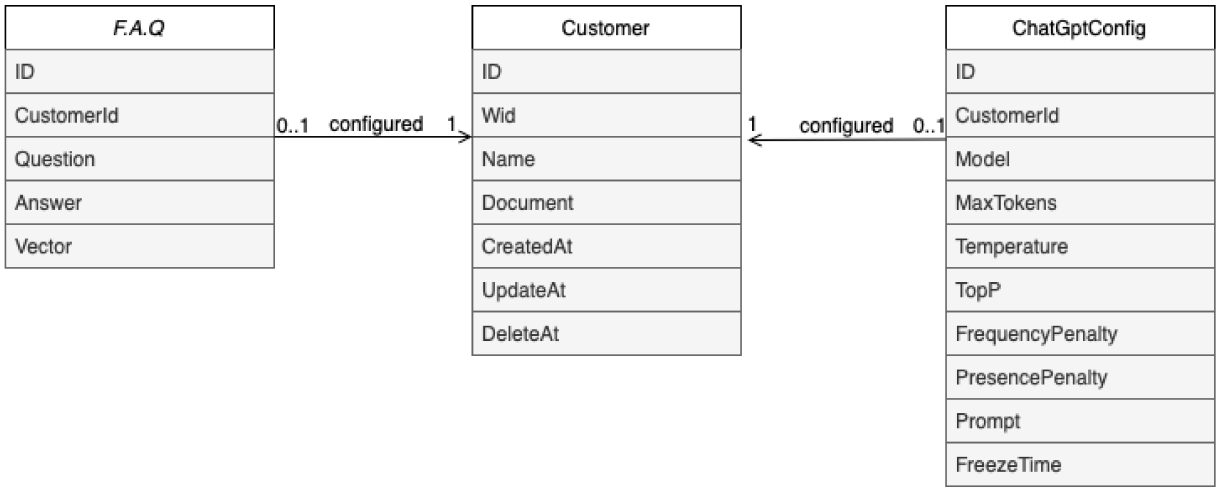
3.1 Diagrama de Classes e Modelos

Neste segmento do estudo, abordaremos a estrutura de classes essenciais que atuam como modelos dentro do sistema. Estas classes são cruciais para o gerenciamento de dados e configurações que permitem a personalização e eficácia da plataforma. As classes principais incluem o Modelo de Cliente, o Modelo de FAQs e o Modelo de Configuração do *GPT*, cada uma desempenhando funções específicas dentro da arquitetura do sistema. Na Figura 4 apresentamos o diagrama de classes, ilustrando as relações e dependências entre essas classes.

Neste diagrama de classes, são apresentadas três classes principais que formam a espinha dorsal do sistema: Modelo de Perguntas e Respostas Frequentes (*FAQ Model*), Modelo de Cliente (*Customer Model*), e Modelo de Configuração do ChatGPT (*ChatGptConfig Model*). Cada uma destas classes é crucial para a funcionalidade e gestão da aplicação, representando diferentes aspectos da interação entre o usuário e o sistema de inteligência artificial.

O Modelo de Cliente serve como o núcleo central, onde cada cliente possui um identificador único e dados pessoais, como nome e documento, além de informações de criação, atualização e exclusão. Este modelo está vinculado às outras duas classes através do campo

Figura 4 – Diagrama de classes



Fonte: Elaborado pelo autor

CustomerId, estabelecendo uma relação de um para muitos (1..n), o que significa que um cliente pode estar associado a várias configurações de *chatbot* e múltiplas entradas de FAQs.

O Modelo de Perguntas e Respostas Frequentes (*FAQ Model*) armazena questões frequentes associadas a cada cliente, contendo campos para identificação, questão, resposta e um vetor que representa um mapeamento matemático da informação. O Modelo de Configuração do *ChatGPT* configura especificações técnicas do modelo de linguagem utilizado, como o tipo de modelo, o máximo de *tokens*, a temperatura, penalidades de frequência e presença, e outras configurações que personalizam a interação da IA conforme as necessidades do cliente.

Cada classe e seus respectivos atributos serão detalhados nas subseções seguintes, fornecendo uma compreensão profunda de como esses componentes interagem e contribuem para a funcionalidade geral do sistema.

3.1.1 Modelo de Cliente (Customer Model)

O Modelo de Cliente encapsula todas as informações de cadastro dos usuários da plataforma. Este modelo foi fundamental para personalizar a experiência do usuário e garantir que as interações sejam adaptadas às necessidades específicas de cada cliente.

Na Figura 5 é representado o Modelo de Cliente estruturado na linguagem *Go* utilizando a biblioteca *Gorm*, a qual é uma peça fundamental para a gestão de dados dentro da plataforma. *Gorm* é um *ORM (Object-Relational Mapping)* para *Go*, que facilita as operações *CRUD* ao abstrair a manipulação direta do banco de dados para uma interface mais intuitiva e segura. Dentro do Modelo de Cliente, cada campo tem uma função específica:

Figura 5 – *Customer Model* em *GO*

```go
package models

import (
    "time"

    "gorm.io/gorm"
)

type Customer struct {
    ID        uint           `json:"id" gorm:"primaryKey"`
    Wid       string         `json:"wid"`
    Name      string         `json:"name"`
    Document  string         `json:"document"`
    CreatedAt time.Time      `json:"created"`
    UpdateAt  time.Time      `json:"updated"`
    DeleteAt  gorm.DeletedAt `gorm:"index" json:"deleted"`
}
```

Fonte: Captura do código fonte do projeto

- ***ID***: Este é o identificador único para cada cliente, geralmente gerado automaticamente para garantir a unicidade.
- ***Wid***: Utilizado para armazenar informações de número telefônico único do *WhatsApp*.
- ***Name***: Armazena o nome do cliente, essencial para personalização das interações.
- ***Document***: Guarda um documento identificador do cliente, como CPF ou CNPJ, o que é crucial para processos de verificação e segurança.
- ***CreatedAt, UpdatedAt, DeleteAt***: São campos de timestamp que registram os momentos de criação, atualização e exclusão do registro no banco de dados, ajudando na manutenção do histórico de alterações e na auditoria de dados.

A utilização do *Gorm* nos modelos do projeto permite uma integração eficiente e robusta com o banco de dados. Por exemplo, *Gorm* suporta automaticamente as operações de migração de esquema, o que simplifica a evolução da base de dados sem intervenção manual significativa. Além disso, com *Gorm*, é possível implementar relações complexas, como relações um-para-muitos ou muitos-para-muitos, de maneira simplificada.

3.1.2 Modelo de Perguntas e Respostas Frequentes (F.A.Q Model)

O Modelo ***FAQ Model***, desempenha uma função vital na gestão de informações de suporte ao cliente oferecendo uma base estruturada para armazenar e recuperar respostas a perguntas comuns. Este modelo é especialmente projetado para facilitar o acesso rápido e eficiente às informações, essencial para manter um alto nível de satisfação do cliente e agilidade no atendimento. A Figura 6 representa a implementação dessa classe em *GO*.

Detalhando mais sobre os atributos do Modelo de Perguntas e Respostas Frequentes:

- ***ID***: Mapeado como um tipo de Universal Unique Identifier (UUID), serve como a chave primária do modelo, garantindo que cada entrada seja única e facilmente acessível. O uso de UUID é uma prática comum em sistemas de informação para gerar identificadores únicos. O uso de UUID é uma prática comum em sistemas de informação para gerar identificadores que são únicos não apenas dentro de um único banco de dados, mas em múltiplos sistemas, aumentando a interoperabilidade e segurança dos dados.
- ***CustomerId***: A conexão entre o *FAQ Model* e o cliente específico através deste campo é crucial. Ela permite que as respostas sejam personalizadas e relevantes para cada usuário, garantindo que o conteúdo do *FAQ Model* seja diretamente aplicável ao contexto do cliente.
- ***Question***: Armazenar a pergunta de forma clara e estruturada facilita a indexação e recuperação de informações. Este campo é fundamental para o funcionamento dos sistemas de busca internos, permitindo que as perguntas sejam encontradas e correspondidas com precisão durante as interações com o cliente.
- ***Answer***: O campo de resposta é igualmente importante, pois contém a informação que será entregue ao cliente. A qualidade e a precisão das respostas armazenadas neste campo podem diretamente impactar a eficácia do suporte fornecido, tornando crucial a manutenção de um alto padrão de conteúdo nas respostas.
- ***Vector***: Este campo representa um mapeamento matemático da informação de pergunta e resposta.

O modelo de *FAQ*, com seus atributos bem definidos, é essencial para manter a organização e a eficiência das operações de atendimento ao cliente. Ele não apenas simplifica a gestão de perguntas frequentes, mas também aprimora a capacidade de resposta e personalização do atendimento, contribuindo significativamente para a satisfação geral do cliente.

Figura 6 – *FAQ Model* em *GO*

```go
package models

import (
    "database/sql/driver"
    "fmt"
    "strconv"
    "strings"

    "github.com/gofrs/uuid"
    "gorm.io/gorm"
)

type Faq struct {
    ID         uuid.UUID `gorm:"primaryKey"`
    CustomerId int       `json:"customer_id"`
    Question   string    `json:"question"`
    Answer     string    `json:"answer"`
    Vector     Vector    `json:"vector,omitempty" gorm:"type:double precision[]"`
}

type Vector []float32
```

Fonte: Captura do código fonte do projeto

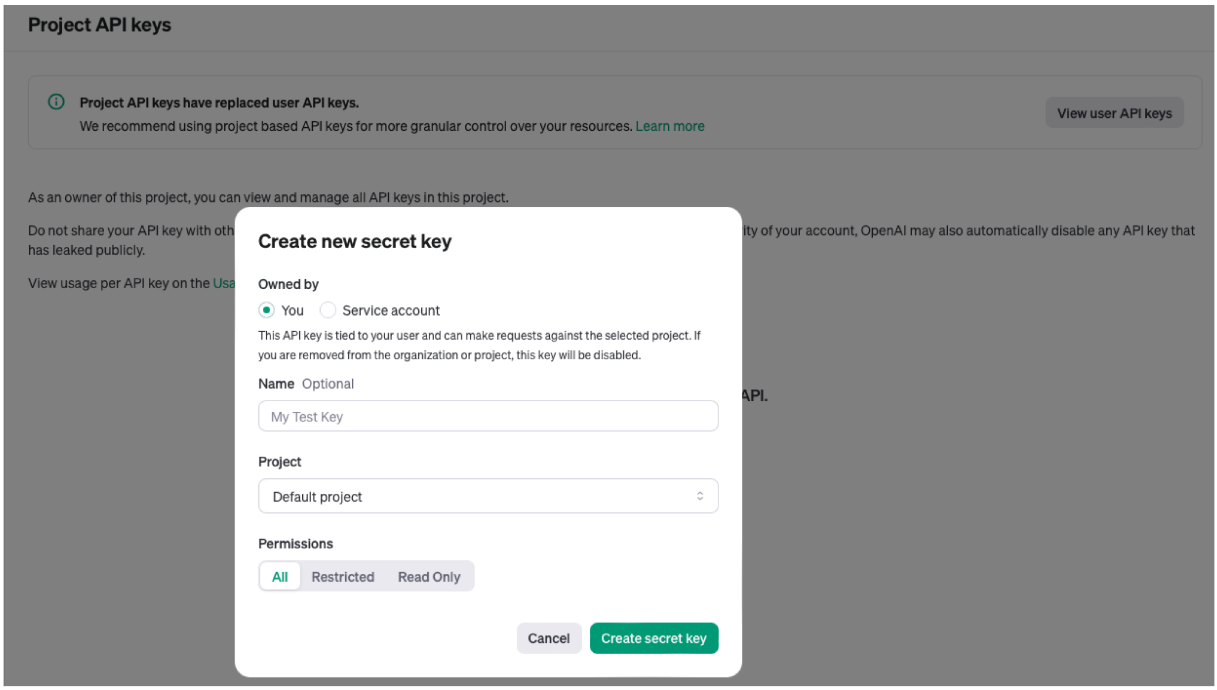
3.1.3 Modelo de Configuração GPT (ChatGPTConfig Model)

O Modelo de Configuração *GPT* abrange as variáveis de configuração necessárias para a personalização da API do *GPT*. Esta classe interage diretamente com o Modelo de Cliente para adaptar as respostas geradas pela IA às particularidades de cada cliente. A API possibilita gerar textos de alta qualidade, realizar tarefas de *completions*, onde a IA é fornecida com um contexto inicial e, com base nisso, continua a gerar texto de forma coerente e contextualizada.

Para começar a utilizar a API do *GPT*, é necessário adquirir uma chave de API. Esta chave atua como um identificador único, vinculando todas as operações realizadas à conta do usuário, permitindo controle de acesso e monitoramento do uso. A Figura 7 representa a interface de criação de chaves na plataforma da API do *GPT*.

A imagem ilustra o início do processo de criação de chaves de API, onde o usuário pode gerar e gerenciar suas chaves, essenciais para autenticar as requisições à API. Uma vez obtida a chave de API, representado por $OPENAI_API_KEY, as requisições podem ser feitas utilizando ferramentas como *cURL*. A seguir, apresentamos exemplos de requisições da própria API para a funcionalidade de *completions* (Figura 8) e a correspondente resposta em *JSON* (Figura 9).

Figura 7 – *GPT API keys*



Fonte: disponível em https://platform.openai.com/api-keys

Figura 8 – *Completion OpenAI API*: exemplo de chamada externa em *cURL*

```
curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
     "model": "gpt-3.5-turbo",
     "messages": [{"role": "user", "content": "Say this is a test!"}],
     "temperature": 0.7
   }'
```

Fonte: disponível em https://platform.openai.com/docs/api-reference/making-requests

Após a requisição ser enviada, a resposta criada pela IA pode ser encontrada na primeira posição do atributo *choices*, especificamente no campo *content* dentro de *messages*. Esse campo contém o texto gerado pela IA com base no contexto fornecido. Além disso, a resposta inclui um campo *role*, que identifica o interlocutor na conversa, podendo ser um dos seguintes:

- ***user***: Representa o usuário na conversa.
- ***system***: Representa o sistema, utilizado para instruções e configurações de contexto.
- ***assistant***: Representa o assistente, que é a IA respondendo ao usuário.

Essa distinção ajuda a organizar a conversa de forma clara, o que é essencial para

Figura 9 – *Completion OpenAI API*: exemplo de resposta em *JSON*

```json
{
    "id": "chatcmpl-abc123",
    "object": "chat.completion",
    "created": 1677858242,
    "model": "gpt-3.5-turbo-0613",
    "usage": {
        "prompt_tokens": 13,
        "completion_tokens": 7,
        "total_tokens": 20
    },
    "choices": [
        {
            "message": {
                "role": "assistant",
                "content": "\n\nThis is a test!"
            },
            "logprobs": null,
            "finish_reason": "stop",
            "index": 0
        }
    ]
}
```

Fonte: disponível em https://platform.openai.com/docs/api-reference/making-requests

criar contextos sequenciais e reativos, como em um *chatbot*. Nesse processo, as instruções gerais de contexto são atribuídas ao *role system*, o histórico das mensagens enviadas pelo usuário é categorizado como *user*, e as respostas geradas pelo assistente ficam sob o *role assistant*. Isso garante que cada parte da interação seja organizada corretamente, facilitando o entendimento e a continuidade da conversa.

O processamento dessas respostas é influenciado significativamente pela configuração dos Modelos de *GPT* utilizados. A escolha do Modelo *GPT*, juntamente com outros parâmetros configuráveis, como *max_tokens*, *temperature*, e *top_p*, determina a qualidade, relevância e variabilidade das respostas geradas. Abaixo, apresentamos uma tabela comparativa dos principais modelos *GPT* disponíveis, destacando suas características de performance, capacidade e custo. A Tabela 1 ajuda a entender melhor as diferenças entre os modelos e a fazer uma escolha informada para o uso no projeto:

A Tabela 1 mostra os principais modelos *GPT* disponíveis, destacando a capacidade

Tabela 1 – Comparação de Modelos GPT

Modelo	Capacidade Máx de Tokens	Velocidade de Resposta	Preço por 1000 tokens
GPT-3.5	4096	Moderada	$0,02
GPT-3.5 Turbo	4096	Alta	$0,002
GPT-4	8192	Moderada	$0,03
GPT-4 Turbo	128k	Alta	$0,015
GPT-4 Turbo Preview	8192	Alta	$0,012

Fonte: Elaborado pelo autor

máxima de *tokens*, a velocidade de resposta e o custo por 1.000 *tokens* processados. O modelo *GPT-4 Turbo* foi escolhido para este projeto devido à sua combinação de alta capacidade e velocidade de resposta, oferecendo uma performance superior ao *GPT-3.5*, a um custo mais acessível que o *GPT-4* tradicional.

O *ChatGPTConfig Model* é crucial para a configuração da API do *GPT*, permitindo ajustes detalhados que influenciam diretamente o comportamento do modelo de linguagem. A Figura 10 representa a implementação do modelo *ChatGPTConfig* em *Go*:

Detalhando mais sobre os atributos do Modelo de Configuração *GPT*:

- ***ID***: Identificador único do registro.
- ***CustomerId***: Referência ao cliente associado a esta configuração.
- ***Model***: Nome do Modelo *GPT* selecionado.
- ***Temperature***: Parâmetro que controla a criatividade das respostas.
- ***TopP***: Probabilidade cumulativa para amostragem.
- ***FrequencyPenalty***: Penalidade para a frequência de novos tokens.
- ***PresencePenalty***: Penalidade para a presença de novos tokens.
- ***Prompt***: Texto inicial fornecido ao modelo para contextualização.
- ***FreezeTime***: Tempo configurado pelo cliente durante o qual o bot ficará inativo após iniciar um atendimento humano.

Esta configuração é crucial para garantir que a interação entre o *chatbot* e os usuários seja não só eficiente mas também segura e adaptada às expectativas do cliente. O atributo *FreezeTime*, em particular, é uma configuração inovadora que permite aos administradores definir um período de descanso para o *bot*, durante o qual o atendimento pode ser continuado ou revisado por um agente humano, assegurando que a tecnologia complementa, mas não substitui completamente, a interação humana. Esta abordagem aumenta a eficácia do serviço e a satisfação do cliente, integrando harmoniosamente a resposta automatizada e o atendimento personalizado.

Cada uma dessas classes é essencial para a arquitetura do sistema, trabalhando em

Figura 10 – *ChatGPTConfig Model* em *GO*

```go
package models

type ChatGPTConfig struct {
    ID               uint    `json:"id" gorm:"primaryKey"`
    Model            string  `json:"model"`
    MaxTokens        int     `json:"max_tokens"`
    Temperature      float32 `json:"temperature"`
    TopP             float32 `json:"top_p"`
    FrequencyPenalty float32 `json:"frequency_penalty"`
    PresencePenalty  float32 `json:"presence_penalty"`
    Prompt           string  `json:"prompt,omitempty"`
    CustomerID       uint    `json:"customer_id,omitempty"`
    FreezeTime       int     `json:"freeze_time,omitempty"`
}
```

Fonte: Captura do código fonte do projeto

conjunto para fornecer uma solução robusta e personalizada para atendimento ao cliente via IAG. A configuração precisa e a integração eficaz desses modelos são fundamentais para o sucesso da implementação e operacionalização da plataforma.

3.2 Algoritmo *F.A.Q*

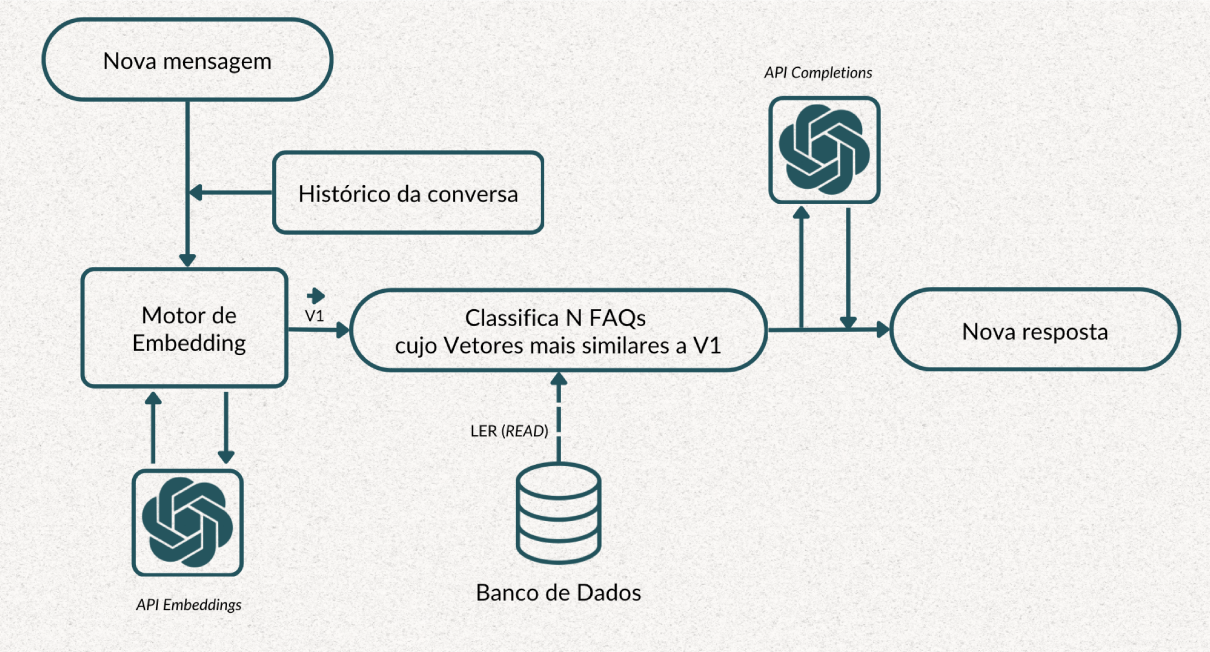
O Algoritmo F.A.Q, nomeado pelo autor, é o algoritmo responsável pela população de dados de contexto no formato de perguntas e respostas associados a um identificador de cliente. Esta estrutura é referenciada como *faqModel*, conforme discutido previamente. O Algoritmo F.A.Q tem a capacidade de:

- Interpretar o atual contexto da interação com o sistema;
- Classificar a pertinência dos *faqModels* com o contexto atual da interação;
- Selecionar os N modelos mais pertinentes para o enriquecimento de contexto para a IA Generativa.

O Algoritmo F.A.Q utiliza uma abstração chamada de *Motor de Embedding*, que gerencia o mapeamento vetorial das informações, utilizando a API da *OpenAI*. Com a capacidade de realizar uma classificação de espaço vetorial, o algoritmo filtra as informações mais relevantes para uma resposta adequada na interação do *chatbot*, diminuindo a quantidade de informação

de contexto enviada para processamento de IA Generativa, poupando recursos e otimizando a performance. A Figura 11 demonstra o fluxograma de como o algoritmo utiliza o *Motor de Embedding* para recuperar os N *faqModels* mais pertinentes para uma resposta contextual:

Figura 11 – Fluxograma do Algoritmo F.A.Q



Fonte: Elaborado pelo autor

1. O algoritmo recupera o atual contexto da interação com o sistema e chama o Motor de Embedding, passando o contexto que retorna um vetor de posição vetorial.
2. Cada *faqModel* cadastrado ao cliente é analisado quanto à aproximação do vetor de posição do contexto atual com o vetor de posição do *faqModel*, utilizando o produto escalar entre os vetores (ou similaridade dos cossenos ordenados de forma decrescente).
3. O Algoritmo captura os N primeiros elementos da consulta.
4. As informações de contexto, compostas pelas interações trocadas e as N perguntas e respostas selecionadas, são enviadas à API de processamento de IA Generativa, retornando uma nova resposta.

Dessa forma, é possível popular diversas informações de contexto para cada cliente sem que haja problemas no desempenho ou elevados custos de processamento, permitindo que o algoritmo forneça respostas precisas na interação com os usuários.

3.2.1 Abstração: Motor de Embedding

O "*Motor de Embedding*"é a abstração responsável pelo mapeamento vetorial das informações de texto parametrizáveis. No projeto, sua implementação é realizada com a API de *Embedding* da *OpenAI*. As Figuras abaixo ilustram como ocorre a interação com essa API para obter *embeddings* de um texto.

Na Figura 12, é apresentado um exemplo de chamada externa à API usando a ferramenta *curl*. O comando em *curl* demonstra como o texto é enviado ao *endpoint* da API de *embeddings* junto com o nome do modelo de *embedding*, como, por exemplo, "*text-embedding-3-small*". A requisição contém o cabeçalho de autorização, que inclui a chave de API, um identificador exclusivo que autentica o usuário e permite o acesso aos recursos da API. Além disso, a requisição também define o conteúdo textual que será processado pela IA. Esse processo permite que a IA transforme o texto em um vetor numérico, conforme será demonstrado na Figura 13.

Figura 12 – *Embedding OpenAI API*: exemplo de chamada externa em curl

Example: Getting embeddings — curl

```
curl https://api.openai.com/v1/embeddings \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "input": "Your text string goes here",
    "model": "text-embedding-3-small"
  }'
```

Fonte: disponível em https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

A Figura 13 apresenta a resposta em formato *JSON* que é retornada pela API de *embeddings* após o envio da requisição. Nesta resposta, o campo data contém o *embedding* propriamente dito, ou seja, uma lista de números flutuantes que representam o texto enviado de forma vetorial. Este vetor numérico é utilizado para identificar padrões semânticos e contextuais no texto, facilitando a recuperação de informações e a análise de similaridade.

Para obter um *embedding*, o texto parametrizável é enviado para o *endpoint* da API de *embeddings* junto com o nome do modelo de *embedding* (por exemplo, *text-embedding-3-small*). A resposta conterá um *embedding* (vetor de números flutuantes), que pode ser extraído, salvo em um banco de dados vetorial para cada *faqModel* ou utilizado para o entendimento de

Figura 13 – *Embedding OpenAI API*: exemplo de resposta em JSON

Example embedding response — json

```json
{
  "object": "list",
  "data": [
    {
      "object": "embedding",
      "index": 0,
      "embedding": [
        -0.006929283495992422,
        -0.005536422007530928,
        ... (omitted for spacing)
        -4.547132266452536e-05,
        -0.024047505110502243
      ],
    }
  ],
  "model": "text-embedding-3-small",
  "usage": {
    "prompt_tokens": 5,
    "total_tokens": 5
  }
}
```

Fonte: disponível em https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

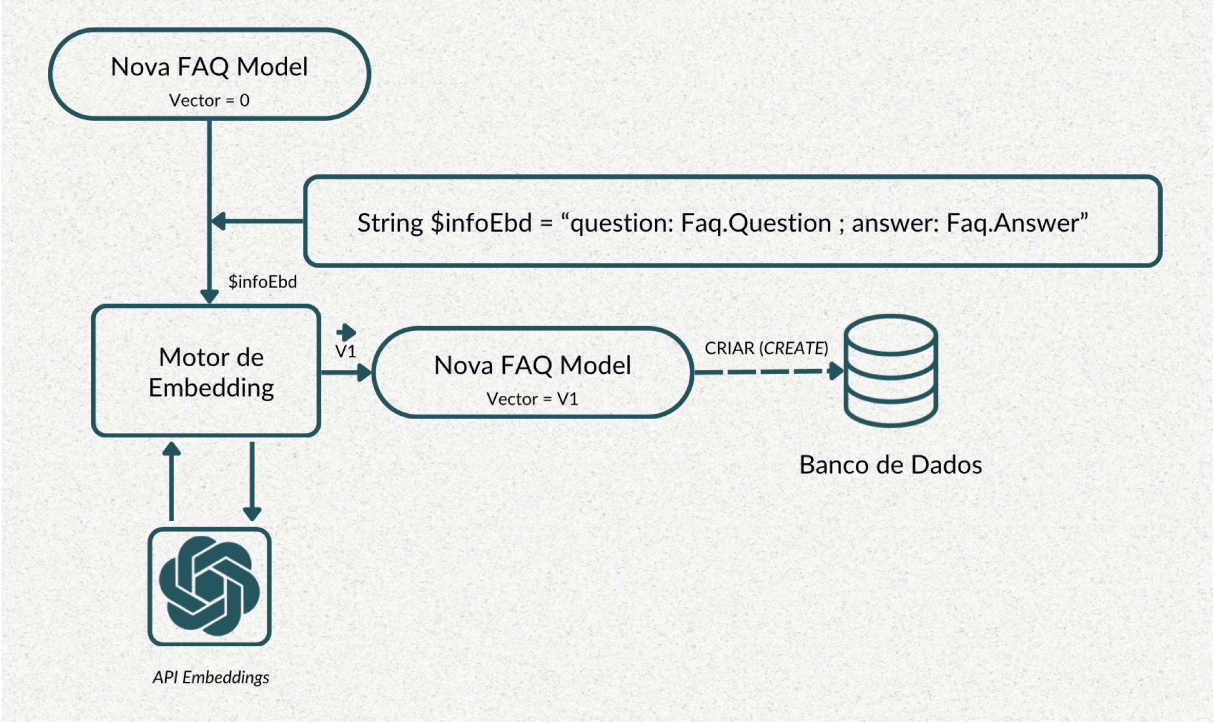
contexto da interação, conforme ilustrado na Figura 11.

A importância de boas práticas de segmentação e abstração ao separar a lógica do *Motor de Embedding* é crucial. Essa abordagem permite tratar o *Motor de Embedding* quase como um serviço independente. Ao isolar essa lógica, o sistema ganha em modularidade, facilitando a manutenção, a escalabilidade e a reutilização do componente em diferentes contextos e aplicações. Além disso, essa prática promove um design mais limpo e organizado, onde cada componente tem responsabilidades bem definidas, contribuindo para a robustez e eficiência do sistema como um todo.

Na Figura 14, pode-se ver um exemplo de aplicabilidade do *Motor de Embedding* no processo de criação de um novo *faqModel*:

1. Ao criar um novo *faqModel*, uma informação do tipo textual é conjecturada: *$infoEbd = "question: faqModel.Question; answer: faqModel.Answer"*.
2. Esta informação *$infoEbd* é enviada para o *Motor de Embedding*, que retorna um valor de mapeamento vetorial.
3. A informação recebida do *Motor de Embedding* é associada ao *faqModel* em criação e, em seguida, salva no banco de dados.

Figura 14 – Processo de criação de uma nova *faqModel*



Fonte: Elaborado pelo autor

Assim, o *Motor de Embedding* possibilita a classificação e recuperação eficiente de informações contextuais relevantes, otimizando a performance e a precisão das respostas fornecidas pela IA Generativa.

3.3 Arquitetura de rede e Multilocação (*Multitenancy*)

A arquitetura é utilizada para gerenciar a comunicação entre o *front-end*, através de instâncias de aplicações de mensageria, como o *WhatsApp*, que interagem diretamente com os usuários, e o *back-end*, que processa as solicitações e gerencia os dados.

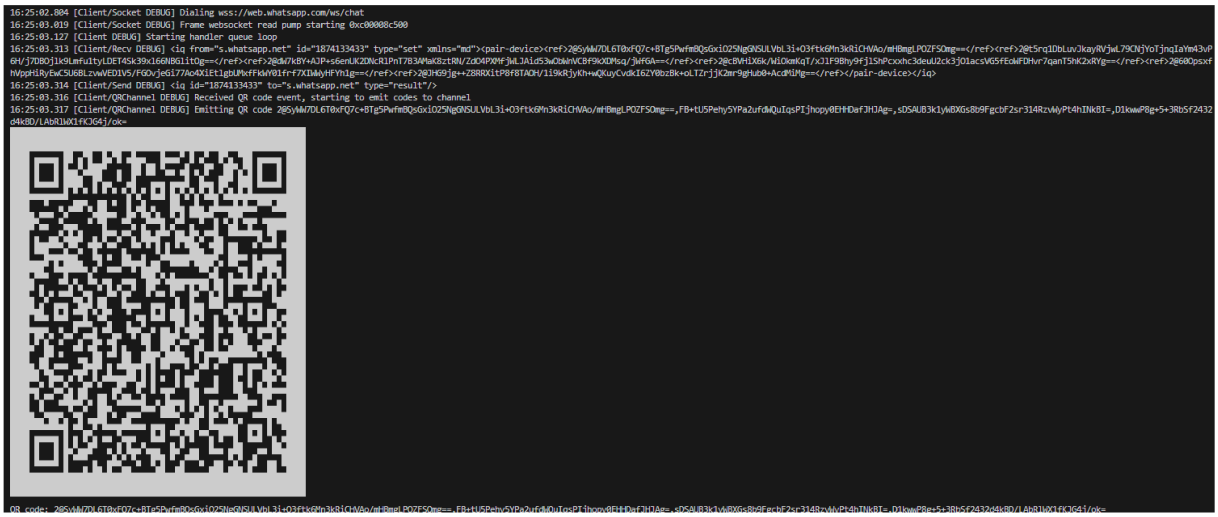
O desempenho na escalabilidade está relacionado à arquitetura proposta possibilitar que novos clientes sejam adicionados ao sistema sem a necessidade de criar novas instâncias de software, utilizando a mesma infraestrutura de forma eficiente. Além disso, a personalização dos serviços é garantida, pois cada cliente pode ter suas próprias configurações e dados, assegurando que as necessidades específicas de cada um sejam atendidas sem interferência de outros clientes.

Esta seção detalha as configurações e implementações essenciais para suportar essa arquitetura, destacando três componentes principais: o Cliente, o *Docker* e o *Heroku*.

3.3.1 Configuração do Cliente

A configuração do cliente utiliza o pacote *Whatsmeow*, que facilita a integração com o *WhatsApp*, permitindo um controle granular sobre o *WhatsApp* vinculado, especialmente no gerenciamento de mensagens recebidas e status. O *Whatsmeow* opera de maneira semelhante ao *WhatsApp Web*, sincronizando informações e criando instâncias independentes para cada cliente através da autenticação via *QR-Code*. A Figura 15 ilustra o processo de autenticação e vinculação da conta *WhatsApp* com o cliente *Whatsmeow*.

Figura 15 – Processo de autenticação do *WhatsApp* com *Whatsmeow*



Fonte: Captura do console executando o cliente

Na Figura 15, o processo de autenticação é iniciado ao escanear o *QR-Code* com o aplicativo do *WhatsApp* no dispositivo móvel. Isso estabelece uma nova sessão, permitindo que as mensagens sejam sincronizadas com a aplicação. Com essa configuração, a aplicação pode mapear e definir o comportamento esperado ao receber novos eventos, como por exemplo, o recebimento de uma nova mensagem que é automaticamente redirecionada ao servidor, onde uma resposta apropriada é gerada e enviada de volta ao usuário.

3.3.2 Configuração do Docker

No projeto, o *Docker* é utilizado para isolar o cliente e o servidor *web* em contêineres separados. A Figura 16 apresenta o arquivo de configuração *DockerFile*, que ilustra o processo de criação de contêineres.

O *DockerFile* mostrado na Figura 16 configura o ambiente necessário para a execução

Figura 16 – Arquivo de configuração Dockerfile do servidor

```
FROM golang

WORKDIR /src
COPY . .
RUN CGO_ENABLED=0 GOOS=linux go build -o whatsapp-api-pv .

FROM alpine:latest AS production
WORKDIR /app

COPY --from=0 /src/whatsapp-api-pv /app
COPY --from=0 /src/server/ssr /app/server/ssr

EXPOSE 5000
CMD ["./whatsapp-api-pv"]
```

Fonte: Captura do código fonte do projeto

do cliente e do servidor *web*. O arquivo começa com a utilização da imagem base do *Golang* e define o diretório de trabalho como `/src`. Em seguida, copia todos os arquivos para o contêiner. O `GOOS=linux` e `CGO_ENABLED=0` são configurados para compilar a aplicação para o ambiente *Linux*. O binário gerado é copiado para um contêiner *Alpine* minimizado, expondo a porta 5000 e definindo o comando de inicialização.

O *DockerFile* configura o ambiente necessário para a execução do cliente e do servidor *web*, garantindo que cada contêiner funcione de maneira independente e segura. Ao isolar o cliente e o servidor em contêineres distintos, é possível realizar atualizações e manutenção em um serviço sem impactar o outro, aumentando a disponibilidade do sistema. Além disso, a utilização de contêineres permite a replicação fácil dos serviços, possibilitando a escalabilidade horizontal conforme a demanda cresce. Dessa forma, cada contêiner contém todas as dependências necessárias, o que assegura consistência entre os ambientes de desenvolvimento, teste e produção, reduzindo o risco de problemas devido a diferenças de configuração.

3.3.3 Configuração do Heroku

A Figura 17 ilustra o arquivo de configuração *heroku.yml*, detalhando a configuração dos *dynos*.

No arquivo *heroku.yml*, cada *dyno* é configurado com identificadores únicos para

Figura 17 – Arquivo de configuração heroku.yml

```yaml
setup:
  addons:
    - plan: heroku-postgresql
      as: DATABASE
build:
  docker:
    web: faq-server/Dockerfile
    cliente_#1: whatsapp-client/Dockerfile
    cliente_#2: whatsapp-client/Dockerfile
    cliente_#3: whatsapp-client/Dockerfile
  config:
    NODE_ENV: production
```

Fonte: Captura do código fonte do projeto

cada cliente, assegurando que os dados de configuração sejam autossuficientes e autocontidos. Esta abordagem não só permite a execução eficiente e escalável de aplicações multilocatárias, mas também garante a integridade e a segurança dos dados de cada cliente, isolando as configurações e personalizações específicas de cada um.

A Figura 17 mostra a configuração do *heroku.yml* onde são especificados os contêineres para o servidor web e para cada cliente individualmente. A seção "*setup*"adiciona o banco de dados *Heroku Postgres*, enquanto a seção "*build*"define os *Dockerfiles* correspondentes ao servidor e a cada cliente (cliente_#1, cliente_#2, cliente_#3). Essa configuração garante que cada cliente tenha seu ambiente isolado e personalizado, promovendo uma operação segura e eficiente.

Em resumo, a configuração acertiva dos componentes Cliente, *Docker* e *Heroku* assegura que o sistema possa gerenciar múltiplos clientes de maneira eficaz. Esta arquitetura permite a escalabilidade horizontal, onde novos clientes podem ser adicionados sem impactar a performance dos existentes, e a personalização dos serviços, atendendo às necessidades específicas de cada cliente.

4 RESULTADOS

Este capítulo apresenta os resultados obtidos com a implementação do sistema de IA Generativa em três empresas de diferentes áreas de atuação: educação, moda e saúde. O objetivo é mostrar como a utilização do *Algoritmo F.A.Q*, abordado na seção 3.2, contribuiu para melhorar o desempenho do atendimento ao cliente, reduzindo o contexto e, consequentemente, os custos operacionais. O algoritmo foi projetado para minimizar a quantidade de informação de contexto enviada para processamento pela IA, diminuindo o consumo de *tokens* e mantendo a coerência nas respostas. Isso é feito de maneira intuitiva, permitindo que perguntas e respostas sejam facilmente adicionadas ao sistema, trazendo benefícios significativos ao gerenciamento da informação.

4.1 Estudo de caso: Empresa A - Educação

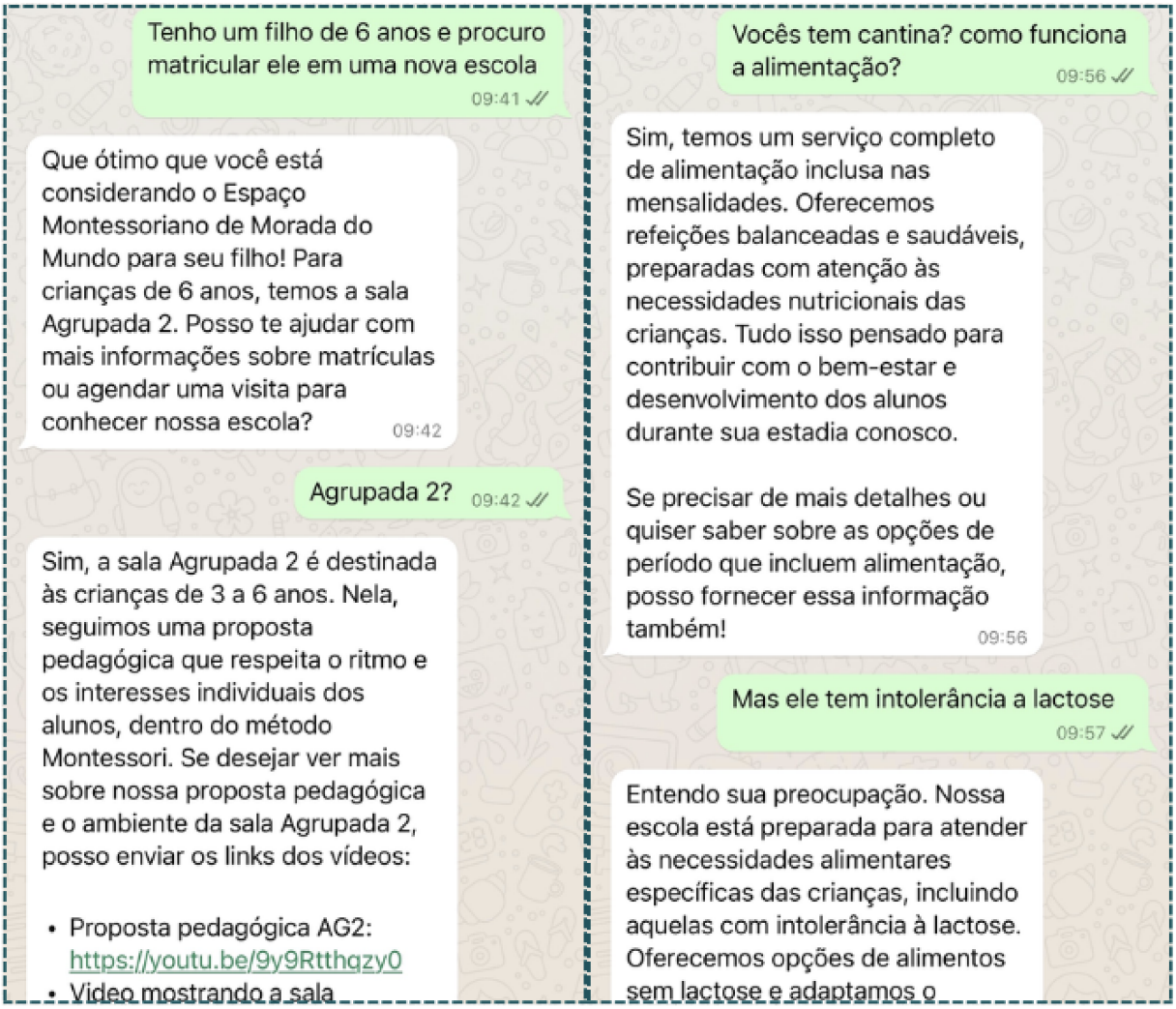
A **Empresa A** é uma escola de educação infantil em Vila Velha - ES que adota a metodologia de ensino Montessoriano. Esse método educativo, desenvolvido pela médica e educadora Maria Montessori, foca no desenvolvimento natural das habilidades da criança, promovendo a autonomia, a liberdade com responsabilidade e o respeito pelo desenvolvimento físico, social e psicológico natural. As crianças aprendem através da interação com materiais didáticos específicos, sem a necessidade de provas ou fardas, agrupadas em salas com idades mistas.

Para atender as necessidades de comunicação com os pais, foi implementado um *chatbot* no *WhatsApp* utilizando o *Algoritmo F.A.Q*, configurado com 46 perguntas e respostas frequentes (*FAQs*) e um tempo de congelamento (*FreezeTime*) de 10 minutos. O *FreezeTime* de 10 minutos é ideal para este cenário, pois permite que o *bot* opere com alta autonomia, respondendo rapidamente às perguntas dos pais e estando ativo nos momentos cruciais de decisão de matrícula.

Os resultados obtidos com a implementação do *chatbot* mostraram uma redução significativa no consumo médio de *tokens*, passando de 7653 *tokens* para 2692 tokens após a aplicação do *Algoritmo F.A.Q*. As interações foram altamente positivas, com o algoritmo entendendo bem as necessidades específicas dos pais e fornecendo respostas detalhadas sobre a metodologia Montessoriana.

A Figura 18 demonstra a interação do usuário com o *bot*. Na primeira interação, o usuário informa que tem um filho de 6 anos e deseja informações sobre matrícula. O *bot*

Figura 18 – Capturas da interação com o *bot* implementado para Empresa A



Fonte: Elaborado pelo autor

responde informando a disponibilidade da agrupada 2, que é destinada a crianças de 3 a 6 anos. Quando o usuário pergunta o que é a agrupada 2, o *bot* explica corretamente, entregando links de vídeos para uma melhor compreensão. Na segunda interação, o usuário menciona que seu filho tem intolerância à lactose, e o *bot* responde compreensivamente, informando sobre a variedade de cardápio disponível que atende a todas as necessidades, inclusive dietas mais restritas.

4.2 Estudo de caso: Empresa B - Moda

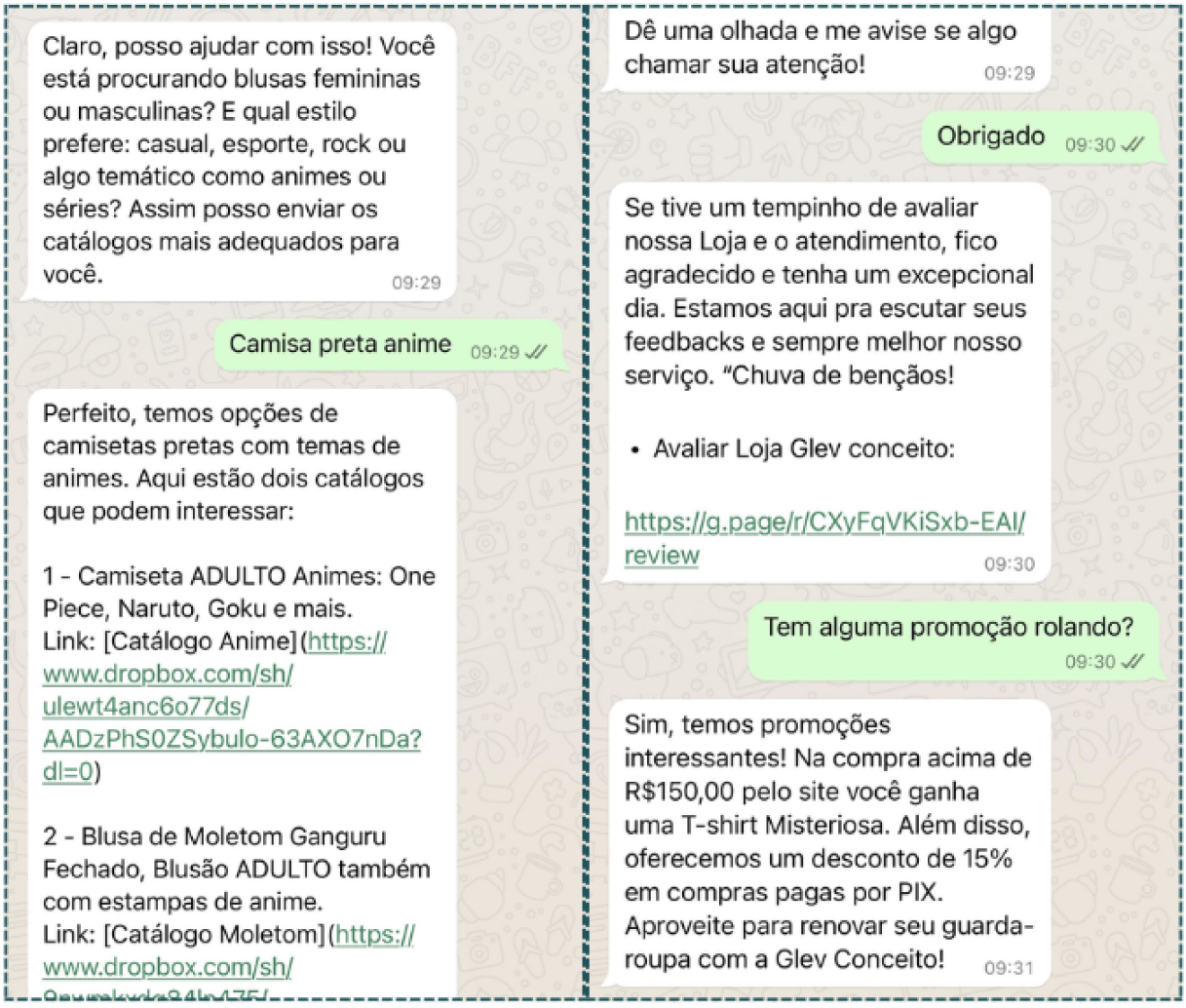
A **Empresa B** é uma loja de roupas em Fortaleza-CE, que vende tanto no atacado quanto no varejo, com presença física e virtual. A IA foi configurada para fornecer links de catálogo segmentado, responder a perguntas sobre promoções, sorteios e facilitar a interação direta com o cliente. O *chatbot* foi configurado com 20 *FAQs* e um *FreezeTime* de 3 minutos. Esta

configuração permite que o *bot* entregue rapidamente links de catálogos e outras informações relevantes, essencial para um ambiente de vendas onde a velocidade de resposta pode influenciar diretamente as vendas.

O algoritmo de segregação e pesquisa dentro do *WhatsApp* garante que os clientes recebam os links de catálogos adequados com base em suas preferências, seja por tipo de roupa ou coleção. O *FreezeTime* de 3 minutos demonstra a alta autonomia do sistema, permitindo respostas rápidas e eficazes, minimizando a necessidade de intervenção humana.

Os resultados obtidos mostraram uma boa redução no consumo médio de *tokens*, passando de 2840 *tokens* para 1698 *tokens* após a aplicação do *Algoritmo F.A.Q*. As interações foram muito positivas, com o *bot* entendendo bem as preferências dos clientes e entregando links precisos para catálogos, além de informações sobre promoções sazonais.

Figura 19 – Capturas da interação com o *bot* implementado para Empresa B



Fonte: Elaborado pelo autor

A Figura 19 demonstra a interação do usuário com o *bot*. Na primeira interação, o usuário expressa interesse em comprar blusas da marca da empresa B. O *bot* busca entender melhor as necessidades do usuário, perguntando sobre estilos específicos. O usuário especifica que deseja camisas de anime, e o *bot* entrega os links dos catálogos com temas de anime, incluindo uma coleção de camisetas e outra de moletons. Na segunda interação, o cliente agradece, e o *bot*, entendendo que a interação está chegando ao fim, pede uma avaliação e envia um link do *Google Empresas* para *feedback*. Ainda nesta interação, o usuário pergunta sobre promoções, e o *bot* informa sobre a promoção ativa no momento.

4.3 Estudo de caso: Empresa C - Saúde

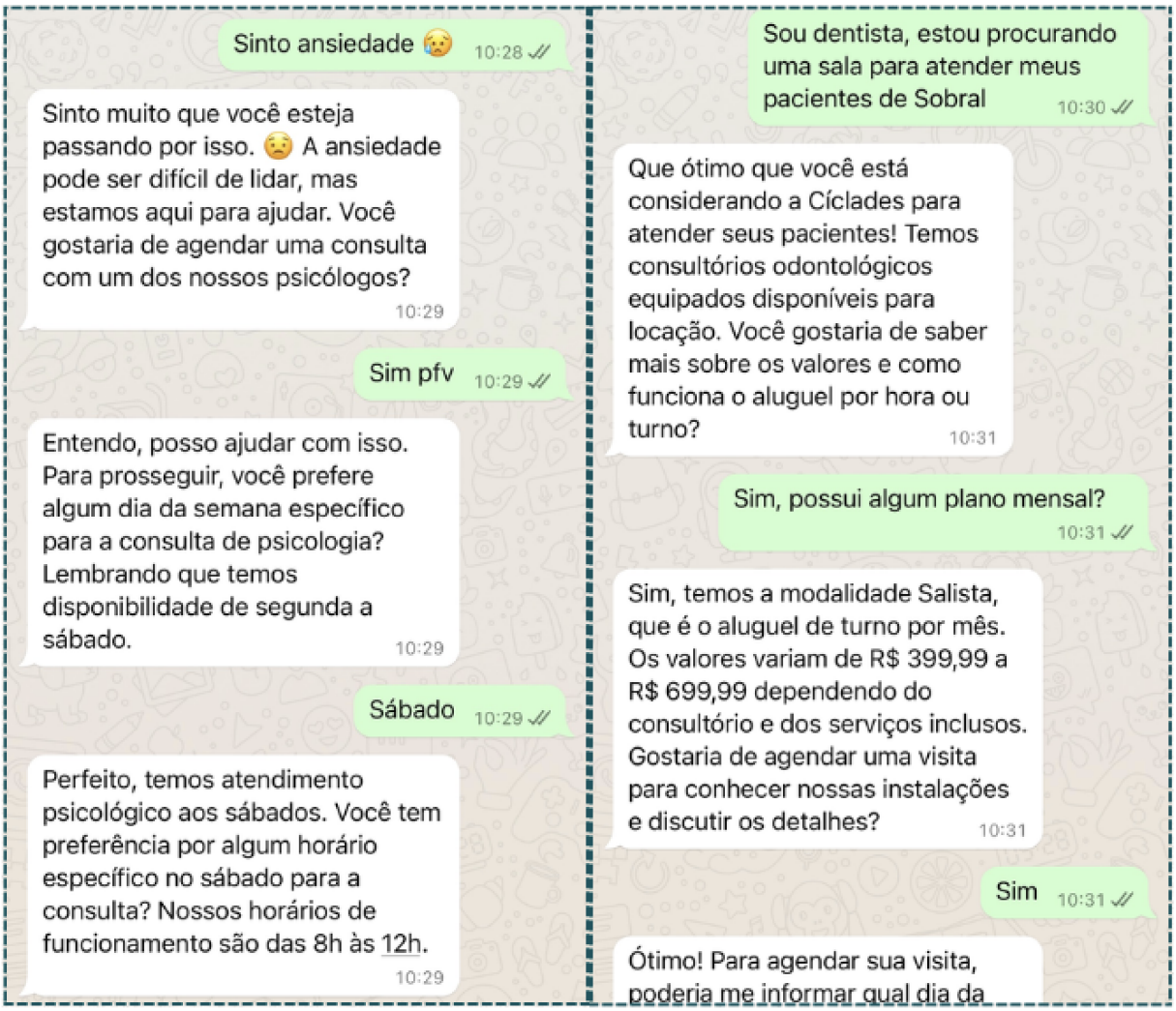
A **Empresa C** é uma clínica de saúde e *coworking* em Sobral-CE. A IA foi implementada para ajudar nas consultas iniciais dos pacientes, fornecer informações sobre convênios, aluguéis de salas e outras questões administrativas. O *chatbot* foi configurado com 28 *FAQs* e um *FreezeTime* de 30 minutos, permitindo respostas detalhadas e a possibilidade de envolver atendimento humano quando necessário. Este tempo de congelamento mais longo é necessário devido à natureza delicada e detalhada das interações clínicas, que frequentemente exigem uma maior participação humana para assegurar a confiança e satisfação do paciente até a marcação de consultas ou visitas.

Os resultados obtidos mostraram uma boa redução no consumo médio de *tokens*, passando de 4625 *tokens* para 2825 *tokens* após a aplicação do *Algoritmo F.A.Q*. As interações foram muito satisfatórias, com o bot respondendo de maneira compreensiva e detalhada, adaptando-se bem tanto a consultas médicas quanto a questões sobre aluguel de salas.

A Figura 20 ilustra a interação do usuário com o *bot*. Na primeira interação, o usuário relata estar sentindo ansiedade, e o *bot*, utilizando uma linguagem reconfortante, pergunta se o usuário deseja marcar uma consulta psicológica. O usuário concorda, e o *bot* prossegue para a confirmação de dia e horário da consulta. Na segunda interação, o usuário se identifica como dentista e procura um consultório para atendimento em Sobral. O *bot* na sequência confirma e pergunta se o usuário deseja saber sobre as modalidades de aluguel de sala. O cliente confirma e pergunta sobre o plano mensal, e o *bot* fornece informações detalhadas sobre essa modalidade, sugerindo um agendamento de visita. O usuário confirma, e em seguida *bot* prossegue para a finalização do agendamento.

A implementação do *Algoritmo F.A.Q* resultou em reduções significativas no con-

Figura 20 – Capturas da interação com o *bot* implementado para Empresa C

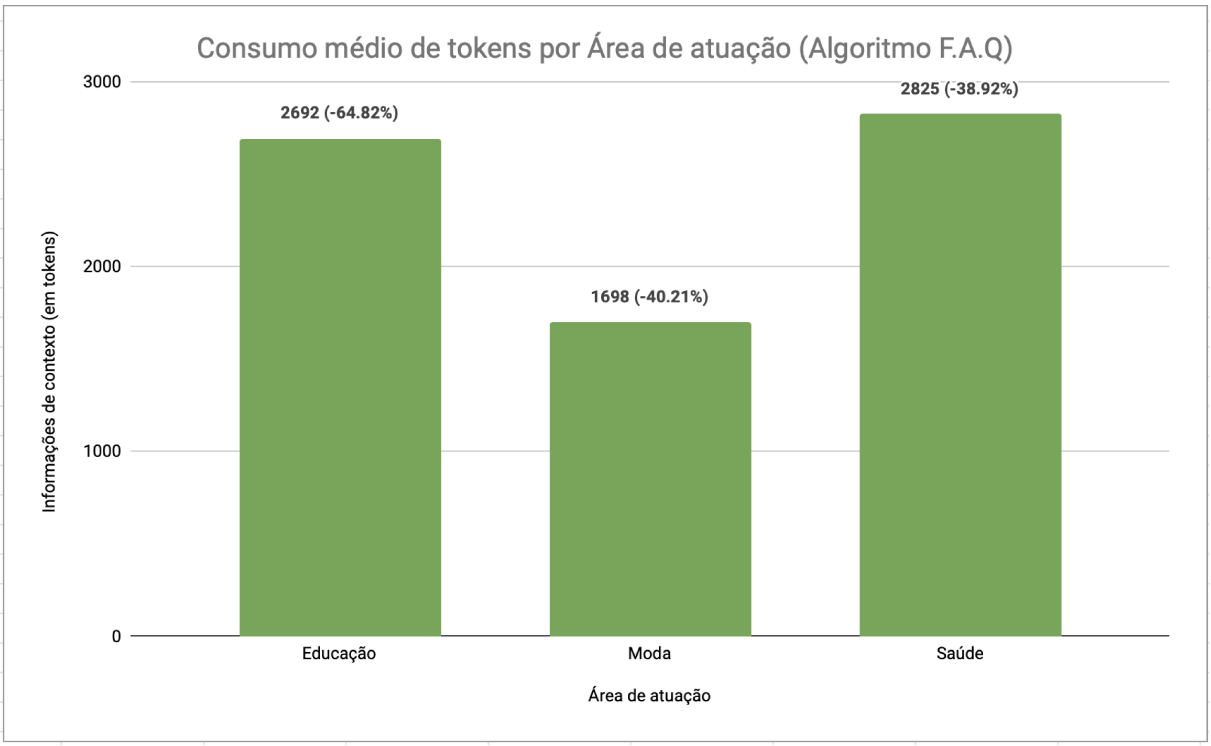


Fonte: Elaborado pelo autor

sumo de *tokens* enviados para IAG, confirmando a eficácia do algoritmo na melhoria da experiência do usuário final. A modelagem das *FAQs* e a quantidade de tokens por *FAQ* influenciaram diretamente nesses resultados, onde quanto mais eficazmente as *FAQs* foram modeladas, maior foi a redução no consumo de *tokens*, minimizando o contexto necessário para responder às consultas dos clientes de forma eficiente.

A Figura 21 apresenta um gráfico comparativo do consumo de *tokens* por interação antes e depois da implementação do algoritmo, destacando as reduções de 7653 para 2692 *tokens* (64.82%) para a **Empresa A**, de 2840 para 1698 *tokens* (40.21%) para a **Empresa B**, e de 4625 para 2825 *tokens* (38.92%) para a **Empresa C**.

Figura 21 – Consumo médio de *tokens* por Área de atuação após *Algoritmo F.A.Q*



Fonte: Elaborado pelo autor

4.4 Análise comparativa e limitações

A implementação do *Algoritmo F.A.Q* nas três empresas analisadas demonstrou resultados variados, dependendo das especificidades de cada setor. As principais variáveis comparadas entre as empresas — eficiência, autonomia e desafios — mostraram na Figura 22 como diferentes configurações do algoritmo podem impactar diretamente a performance do sistema.

Figura 22 – Quadro comparativo entre as empresas analisadas

Empresa	Área de atuação	Quantidade de Faqs	FreezeTime	Eficiência do Sistema	Autonomia do Sistema	Desafios de Implementação
Empresa A	Educação	46	10 minutos	Alta: Redução de 64.82% no consumo de tokens	Boa autonomia com rápida resposta e continuidade.	Relativamente simples, atendimento focado em informações e matrícula.
Empresa B	Moda	20	3 minutos	Moderada: Redução de 40.21% no consumo de tokens	Muito alta: Sistema independente para fornecer links de catálogo de forma rápida.	Desafios mínimos: Processos rápidos e transacionais, foco na entrega de links.
Empresa C	Saúde	28	30 minutos	Moderada: Redução de 38.92% no consumo de tokens	Moderada: Necessidade de interação humana prolongada.	Desafiador: Atendimentos mais complexos que envolvem consulta e aluguel de salas.

Fonte: Elaborado pelo autor

A eficiência do sistema foi um dos aspectos mais notáveis ao se avaliar a redução do

consumo de *tokens*. A **Empresa A**, no setor de educação, apresentou uma eficiência alta, com uma redução de 64,82% no consumo de *tokens*. Isso se deve à configuração otimizada de 46 FAQs e um *FreezeTime* de 10 minutos, que garantiu respostas rápidas e precisas para os usuários. A escolha desse *FreezeTime* permitiu um equilíbrio entre a autonomia do *chatbot* e a necessidade de supervisão humana ocasional, mantendo o atendimento eficiente e reduzindo a sobrecarga do sistema.

Por outro lado, a **Empresa B**, no setor de vendas, obteve uma eficiência moderada, com uma redução de 40,21% no consumo de *tokens*. Esse resultado está diretamente relacionado ao *FreezeTime* mais curto (3 minutos), que favoreceu um atendimento mais rápido e transacional, essencial para um ambiente de vendas onde a agilidade nas respostas é um fator crítico. A configuração de 20 FAQs foi suficiente para responder a perguntas frequentes, mas a redução no consumo de *tokens* foi inferior à da Empresa A, devido à natureza das interações, que exigiam menos contexto e uma maior rotatividade de consultas.

A **Empresa C**, no setor de saúde, apresentou a menor eficiência, com uma redução de 38,92% no consumo de *tokens*. A complexidade do atendimento clínico, que demanda maior detalhamento nas interações, fez com que o sistema exigisse mais contexto para fornecer respostas precisas. Além disso, o *FreezeTime* de 30 minutos, necessário para garantir um atendimento pessoal satisfatório, limitou a eficiência em termos de redução de *tokens*. Isso reflete a necessidade de maior intervenção humana e uma interação mais cuidadosa no setor de saúde.

A autonomia do *Algoritmo F.A.Q* variou significativamente entre as empresas, dependendo do tipo de atendimento e da configuração do *FreezeTime*. Na **Empresa A**, o *chatbot* operou com alta autonomia, sendo capaz de responder rapidamente a perguntas frequentes e fornecer informações detalhadas sobre o sistema de ensino. A configuração de 10 minutos de *FreezeTime* foi crucial para garantir que o *bot* pudesse atuar sem a necessidade constante de supervisão humana, oferecendo respostas precisas e consistentes, o que aumentou a confiança dos usuários.

Na **Empresa B**, a autonomia também foi alta, mas com uma abordagem mais voltada para o comércio, onde o *bot* foi eficaz em fornecer links de produtos e informações sobre promoções de forma rápida. O *FreezeTime* de 3 minutos permitiu uma resposta imediata às necessidades dos clientes, sem a necessidade de intervenções humanas constantes, o que foi essencial para o dinamismo do setor de vendas.

Por outro lado, a **Empresa C** exigiu um *FreezeTime* maior (30 minutos), o que

resultou em menor autonomia do sistema. A necessidade de um atendimento mais detalhado e sensível, especialmente em interações relacionadas à saúde, fez com que fosse necessário um período maior para respostas e, muitas vezes, a participação humana se fez essencial para garantir a precisão e o cuidado no atendimento. Isso limitou a capacidade do *chatbot* de operar de forma completamente autônoma, mas garantiu uma maior confiança dos pacientes nas respostas recebidas.

A implementação do *Algoritmo F.A.Q* enfrentou diferentes desafios em cada uma das empresas. Na **Empresa A**, o principal desafio foi a adaptação do *chatbot* ao contexto educacional, onde os pais e responsáveis buscavam informações detalhadas sobre a metodologia de ensino e o sistema escolar. Ajustes contínuos no início do processo foram necessários para garantir que o *chatbot* entendesse o contexto das interações.

Na **Empresa B**, o desafio foi encontrar o equilíbrio certo entre o número de FAQs e a velocidade de resposta. O foco nas vendas e a necessidade de um atendimento rápido exigiram um *chatbot* altamente ágil, o que demandou ajustes na configuração de *FreezeTime* e na categorização das perguntas frequentes. A complexidade das interações, que envolviam questões sobre produtos e promoções, exigiu uma adaptação constante para manter a relevância das respostas.

Na **Empresa C**, o desafio mais significativo foi lidar com a complexidade das interações clínicas. O atendimento na área da saúde é naturalmente mais detalhado e exige maior cuidado nas respostas. Isso levou a um maior período de ajustes na implementação do *Algoritmo F.A.Q*, especialmente em questões mais sensíveis como agendamentos médicos e dúvidas sobre convênios. Além disso, o treinamento dos operadores humanos para colaborar de forma eficiente com o *chatbot* foi mais demorado, o que impactou inicialmente na fluidez do atendimento.

Em suma, as três empresas apresentaram diferentes níveis de eficiência, autonomia e desafios com a implementação do *Algoritmo F.A.Q*, refletindo as especificidades de cada setor. O algoritmo se mostrou flexível, permitindo ajustes e configurações que atenderam às necessidades de cada contexto, mas as limitações, especialmente no setor de saúde, demonstram que a interação humana ainda é necessária para garantir a qualidade do atendimento em situações mais complexas.

4.5 Considerações finais

Este trabalho explorou o uso da IA Generativa no suporte ao cliente por meio da implementação do *Algoritmo F.A.Q*. Os estudos de caso realizados com empresas de diferentes setores — educação, moda e saúde — demonstraram a eficácia do algoritmo em reduzir significativamente o consumo de *tokens*, o que resultou em economia de recursos computacionais e melhorou a eficiência operacional do sistema.

Os resultados obtidos destacaram a flexibilidade e a adaptabilidade do *Algoritmo F.A.Q* para diferentes cenários de negócios. Empresas com demandas mais transacionais, como a **Empresa B** (moda), se beneficiaram de uma implementação com *FreezeTime* mais curto, otimizando o tempo de resposta e a agilidade no atendimento. Por outro lado, setores como o da saúde (**Empresa C**) apresentaram desafios mais complexos devido à necessidade de maior detalhamento nas interações, o que exigiu uma configuração específica para equilibrar a automação e a intervenção humana.

A análise comparativa entre os estudos de caso também ressaltou a importância da personalização na implementação de soluções de IA Generativa. O ajuste adequado de parâmetros, como a quantidade de *FAQs* e o *FreezeTime*, mostrou ser fundamental para garantir a autonomia do sistema sem comprometer a qualidade das respostas. Esses ajustes também influenciaram diretamente a redução do consumo de *tokens*, evidenciando o potencial de economia de custos operacionais.

Além disso, o *Algoritmo F.A.Q* comprovou sua eficácia ao facilitar a personalização das interações com os usuários de maneira intuitiva. A capacidade de adicionar e ajustar perguntas e respostas conforme as necessidades do negócio garante que a solução permaneça relevante e escalável ao longo do tempo, adaptando-se a diferentes setores e demandas de clientes.

As contribuições deste estudo são diversas. Primeiro, ele oferece um guia prático para empresas que desejam implementar IA Generativa em sistemas de atendimento ao cliente, proporcionando não apenas eficiência, mas também insights sobre a gestão de *tokens* e personalização de interações. Segundo, valida a aplicabilidade de soluções baseadas em *FAQs* em setores variados, demonstrando que o uso de IA pode ir além de simples automação, sendo capaz de personalizar e melhorar o atendimento.

Por fim, o estudo confirma que, mesmo em setores onde a intervenção humana é necessária, como o da saúde, o uso combinado de IA Generativa com operadores humanos pode proporcionar ganhos operacionais substanciais, mantendo a qualidade no atendimento.

5 CONCLUSÃO

O objetivo principal deste estudo foi analisar a evolução da IA Generativa, com foco na gestão de *tokens* e na implementação de soluções de IAG no suporte ao cliente. Ao longo dos capítulos, foram explorados os desafios e as soluções para otimizar o desempenho de sistemas baseados em IA, especialmente por meio do *Algoritmo F.A.Q*. Embora os testes realizados no ambiente de desenvolvimento não tenham fornecido resultados conclusivos, é importante destacar que a verdadeira eficiência do *Algoritmo F.A.Q* foi observada em ambientes de produção, onde sua aplicação demonstrou impacto real. A eficiência do algoritmo está intimamente ligada à sua capacidade de reduzir a quantidade de *tokens* enviados para processamento sem comprometer a qualidade das respostas, o que resulta em um uso otimizado de recursos.

Os objetivos estabelecidos no início do trabalho foram amplamente alcançados. Foi possível não apenas explorar a evolução tecnológica da IAG, mas também identificar e validar estratégias eficazes para a gestão de *tokens*, atingindo a finalidade principal do estudo: desenvolver um algoritmo capaz de reduzir o consumo de *tokens* sem perda de qualidade no atendimento ao cliente. O *Algoritmo F.A.Q*, ao integrar um mecanismo inteligente de redução de contexto, mostrou-se capaz de otimizar o fluxo de informação, mantendo o nível adequado de respostas, tanto no ambiente de produção quanto nos testes preliminares.

Entre as principais contribuições deste estudo estão a implementação de um sistema de automatização de atendimento com IAG utilizando tecnologias avançadas, como o *ChatGPT*, integrado a uma linguagem de alto desempenho como *Go*. A arquitetura *multitenant* e o uso de *API REST* proporcionaram um ambiente robusto e escalável, capaz de atender múltiplos clientes de forma eficiente. Este trabalho também serve como guia para futuros projetos, oferecendo uma base sólida para a implementação de sistemas inteligentes que gerenciam recursos de maneira eficiente e escalável.

5.1 Sugestões para Trabalhos Futuros

Futuros trabalhos podem focar no desenvolvimento de novas heurísticas que otimizem a seleção de contexto. Uma outra sugestão seria traçar um estudo comparativo de consumo médio para diferentes modelos de IA, ou a integração de funções externas, como um sistema de avaliação do *bot*, são áreas promissoras para pesquisa.

A partir dos resultados obtidos, recomenda-se a continuidade da coleta e análise de

dados para um entendimento mais profundo e assertivo do impacto da tecnologia implementada. Essa análise poderá ser útil para identificar padrões de comportamento em diferentes setores e otimizar ainda mais as estratégias de uso da IAG em contextos empresariais variados. Além disso, a validação externa das tecnologias propostas poderá fornecer *insights* valiosos para o aprimoramento contínuo da solução.

O Apêndice A contém uma consultoria gerencial feita em parceria com o Sebrae, que fundamenta o potencial da ferramenta em questão. Este relatório traz um estudo detalhado sobre o mercado e as estratégias de crescimento da *Iabots*, sugerindo práticas de escalabilidade e aprimoramento tecnológico para futuros desenvolvimentos.

Este trabalho demonstrou o potencial da IA Generativa no suporte ao cliente, destacando a eficácia do *Algoritmo F.A.Q*. A evolução contínua da tecnologia promete avanços significativos, abrindo novas possibilidades para a personalização e eficiência no atendimento ao cliente. A implementação prática e a validação dos resultados com usuários reais comprovam a viabilidade das soluções propostas, incentivando a continuidade dos estudos na área e a aplicação prática das tecnologias emergentes.

REFERÊNCIAS

ALVES, A. Introdução à api da openai. **SBC Livros**, 2023. Acesso em: 13 mar. 2024. Disponível em: <https://sol.sbc.org.br/livros/index.php/sbc/catalog/download/129/559/864?inline=1>.

AWARI, I. Arquitetura de software cliente-servidor: Estrutura e interação entre os componentes. **Awari**, 2023. Acesso em: 5 mar. 2024. Disponível em: <https://awari.com.br/arquitetura-de-software-cliente-servidor-estrutura-e-interacao-entre-os-componentes/>.

BEAT, V. Chatbots vão responder por 70% das interações com clientes até 2022. **NewVoice**, 05 2021. Acesso em: 02 nov. 2023. Disponível em: <https://newvoice.ai/2021/05/21/chatbots-vao-responder-por-70-das-interacoes-com-clientes-ate-2022/>.

CLARK, J. O que é o heroku? **Back4App Blog**, 2023. Acesso em: 5 mar. 2024. Disponível em: <https://blog.back4app.com/pt/o-que-e-o-heroku/>.

DUARTE, A. R. Oportunidades, riscos e desafios do chatgpt para a comunicação empresarial. **The Trends Hub**, n. 3, 06 2023. Acesso em: 01 nov. 2023. Disponível em: <https://parc.ipp.pt/index.php/trendshub/article/view/5043>.

FAKHRUDDIN, F. Demystifying containers: Step by step guide to docker on heroku. **AppCloud101**, 2016. Acesso em: 6 mar. 2024. Disponível em: <https://www.appcloud101.com/test-2/>.

GEORGE, A. S.; GEORGE, A.; MARTIN, A. A review of chatgpt ai's impact on several business sectors. **Partners Universal International Innovation Journal (PUIIJ)**, v. 01, n. 01, 02 2023. Disponível em: <https://doi.org/10.5281/zenodo.7644359>.

HOHENSTEIN, J.; KIZILCEC, R. F.; DIFRANZO, D.; AGHAJARI, Z.; MIECZKOWSKI, H.; LEVY, K.; NAAMAN, M.; HANCOCK, J.; JUNG, M. F. Artificial intelligence in communication impacts language and social relationships. **Scientific Reports**, v. 01, n. 01, 04 2023. Acesso em: 05 nov. 2023. Disponível em: <https://doi.org/10.1038/s41598-023-30938-9>.

IBM, I. O que é o docker? **IBM**, 2023. Acesso em: 5 mar. 2024. Disponível em: <https://www.ibm.com/br-pt/topics/docker>.

IMAGINEDONE, O. O que é o heroku e como a ferramenta revolucionou o desenvolvimento escalável? **Imaginedone**, 2023. Acesso em: 6 mar. 2024. Disponível em: <https://imaginedone.com.br/artigos/o-que-e-o-heroku/>.

LAM, L. Chatbots são tendência no instagram e impactam no aumento das vendas. **Olhar Digital**, 03 2022. Acesso em: 05 nov. 2023. Disponível em: <https://olhardigital.com.br/2022/03/11/pro/chatbots-sao-tendencia-no-instagram-e-impactam-no-aumento-das-vendas/>.

MAGALHAES, M. R.; CASTRO, R. B. d. Avaliação do uso de chatbots por parte das empresas como meio de atendimento ao consumidor. **COPPE UFRJ**, 03 2019. Acesso em: 01 nov. 2023.

NASCIMENTO, T. Vendas do comércio eletrônico têm alta de 73,88% em 2020. **Estadão**, 02 2021. Acesso em: 05 nov. 2023. Disponível em: <https://economia.uol.com.br/noticias/estadao-conteudo/2021/02/01/vendas-do-comercio-eletronico-tem-alta-de-7388-em-2020-mostra-indice.htm>.

OPENAI, I. Api reference: Embeddings. **OpenAI API Documentation**, 2023. Acesso em: 13 mar. 2024. Disponível em: <https://platform.openai.com/docs/api-reference/embeddings>.

PEREIRA, E. Arquitetura multi-tenancy. **Medium**, 2019. Acesso em: 7 mar. 2024. Disponível em: <https://medium.com/@edytarcio/arquitetura-multi-tenancy-bb7b47d7ba>.

PLATFORMS, I. M. Meta content library api version v2.0. **Meta Content Library**, v. 02, 2024. Acesso em: 05 jan. 2024. Disponível em: <https://doi.org/10.48680/meta.metacontentlibraryapi.2.0>.

POSTGRESQL, O. What is postgresql? **PostgreSQL Documentation**, v. 02, 2023. Acesso em: 05 jan. 2024. Disponível em: <https://www.postgresql.org/docs/current/intro-whatis.html>.

STARTUPI, I. Startup cearense que deseja transformar a comunicação com chatbots recebe aporte da kptl. **Startupi**, 06 2021. Acesso em: 01 nov. 2023. Disponível em: <https://startupi.com.br/2021/06/startup-que-deseja-transformar-a-comunicacao-com-chatbots-recebe-aporte-da-kptl/>.

TAKEBLIP, I. Quem somos. **TakeBlip**, 08 2020. Acesso em: 01 nov. 2023. Disponível em: <https://www.salesforce.com/content/dam/web/it_it/www/pdf/connected-customer-report-2020.pdf>.

APÊNDICE A – MENTORIA DE IDEAÇÃO E VALIDAÇÃO: IABOTS

RELATÓRIO FINAL
CONSULTORIA GERENCIAL

NOME DA CONSULTORIA: Mentoria de Ideação e Validação
RAZÃO SOCIAL (CLIENTE): Ítalo Carvalho Teixeira Barros CNPJ: 42.447.636/0001-26
PRESTADOR DO SERVIÇO: SANTORINI CONSULTORIA E TREINAMENTO LTDA
PERÍODO DE EXECUÇÃO: 28/10/2023 a 30/11/2023
ESCRITÓRIO REGIONAL/UNIDADE: SOBRAL CE
GESTOR RESPONSÁVEL (SEBRAE): MARIANA LINHARES CAVALCANTE DIAS

Startup: IABOTs

Fundador: Ítalo Carvalho

Tempo: 4 horas

Area de atuação: Chatbot com IA que pode ser adaptado para EDTECH, HEALTHTECH, HOSPITALITYTECH, RETAILTECH, entre outras.

1. INTRODUÇÃO – SITUAÇÃO INICIAL

A startup IABOTs, idealizada por Ítalo, um promissor graduando da Universidade Federal do Ceará, representa um excelente exemplo de como a inteligência artificial pode ser aplicada ao atendimento ao cliente. A proposta central da empresa é oferecer uma solução baseada em CHAT GPT, customizável às necessidades específicas de cada cliente. A ferramenta tem potencial para revolucionar o atendimento em diversos setores, como comércio, educação, hotelaria e saúde, programando-se para responder perguntas frequentes e fornecer informações detalhadas com a eficiência de um atendente humano.

2. DESENVOLVIMENTO – SOLUÇÕES IMPLEMENTADAS

ATIVIDADE	Estratégia Comercial e Educação de Mercado
DESCRIÇÃO	Durante a mentoria conversamos sobre a resistência do mercado à substituição de funcionários por IA é uma barreira significativa. Para superá-la, a IABOTs deve investir em uma estratégia educacional, demonstrando não apenas a eficácia da ferramenta, mas também como ela pode coexistir e potencializar o trabalho humano. Isso pode ser feito através de estudos de caso, demonstrações ao vivo e webinars educativos, que ilustrem o retorno sobre o investimento (ROI) e a melhoria na satisfação do cliente. Além disso, é essencial construir uma estratégia de marketing digital que posicione a IABOTs como líder em inovação no atendimento ao cliente. Isso inclui a criação de conteúdo relevante, uso de SEO, marketing de influência e presença ativa nas redes sociais, focando em setores específicos e criando mensagens que ressoem com as dores e desejos de cada público-alvo.

ATIVIDADE	Suporte Técnico e Escalabilidade
DESCRIÇÃO	Em um segundo momento de mentoria abordamos que inicialmente, é crucial para a IABOTs garantir que sua plataforma seja robusta o suficiente para suportar um alto volume de usuários simultâneos. Isso passa pela escolha de uma infraestrutura de servidores escalável e confiável, que possa se expandir conforme a demanda aumenta. Soluções em nuvem como AWS, Google Cloud ou Azure oferecem serviços que podem ser ajustados automaticamente para atender às necessidades de tráfego, garantindo disponibilidade e desempenho. Neste contexto, recomenda-se a realização de testes de carga e estresse para prever o comportamento da plataforma em diferentes cenários de uso. Parcerias com instituições acadêmicas podem ser úteis para conduzir esses testes, oferecendo uma oportunidade de colaboração com a UFC, por exemplo, que poderia beneficiar tanto os estudantes quanto a startup.

3. CONCLUSÃO E SUGESTÕES DE ATIVIDADES FUTURAS:

A IABOTs tem em mãos uma tecnologia promissora com aplicabilidade vasta e potencial de crescimento exponencial. Entretanto, para alcançar o sucesso, é necessário superar desafios técnicos relacionados à escalabilidade e desafios de mercado relacionados à aceitação da tecnologia.

Portanto, sugerimos como medidas futuras:

- Implementação de uma infraestrutura de TI escalável, com o apoio de testes de carga.
- Desenvolvimento de um programa educativo para o mercado, mostrando os benefícios e a complementaridade da IA no trabalho humano.
- Criação de uma estratégia de marketing segmentada e baseada em conteúdo de valor para educar o mercado e promover a ferramenta.
- Estabelecimento de parcerias estratégicas com empresas e instituições educacionais para validação e divulgação da tecnologia.
- Monitoramento contínuo e ajuste de estratégias conforme feedback e dados de uso da plataforma.

Concluímos que a IABOTs está posicionada para ser uma pioneira na área de atendimento automatizado, desde que navegue com cuidado e estratégia pelo atual panorama tecnológico e de mercado.

4. REGISTROS FOTOGRÁFICOS