

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SÃO PAULO**

Murilo Prado Manfredini

cj3035913

TÍTULO DO TRABALHO

Sistema para Chatbot integrado com IA

CAMPOS DO JORDÃO

2025

RESUMO

Este projeto teve como foco o desenvolvimento de uma estrutura de banco de dados destinada a apoiar um sistema de atendimento automatizado via WhatsApp, integrado a um fluxo que utiliza estruturas de agentes de IA para interpretações e escrita de respostas utilizando conceitos de engenharia de contexto. A solução proposta buscou atender a quatro requisitos principais: (1) registro e gerenciamento de informações de clientes; (2) armazenamento de embeddings utilizados pelo mecanismo de leitura e compreensão da IA; (3) organização do histórico de conversas (chat_histories) para fins de acompanhamento e melhoria do atendimento; (4) disponibilização de indicadores e métricas essenciais para a construção de dashboards analíticos. A implementação desses componentes visa garantir um funcionamento eficiente do sistema, proporcionar melhor rastreabilidade das interações e oferecer subsídios para análises estratégicas com base nos dados coletados.

Palavras-Chave: Inteligência Artificial, chatbot, engenharia de contexto

ABSTRACT

This project focused on developing a database structure designed to support an automated WhatsApp service system, integrated with a workflow that uses AI agent frameworks for interpreting and generating responses based on context-engineering concepts. The proposed solution aimed to meet four main requirements: (1) recording and managing customer information; (2) storing embeddings used by the AI reading and

comprehension mechanism; (3) organizing conversation history (chat_histories) for follow-up and service improvement; and (4) providing key indicators and metrics for building analytical dashboards. The implementation of these components aims to ensure efficient system operation, improve traceability of interactions, and provide insights for strategic analyses based on the collected data.

Keywords: Artificial Intelligence, chatbot, context engineering

SUMÁRIO

1	INTRODUÇÃO	2
2	METODOLOGIA	3
3	RESULTADOS OBTIDOS	4
4	CONCLUSÃO	5
5	REFERÊNCIAS BIBLIOGRÁFICAS	6

INTRODUÇÃO

O setor de hotelaria depende fortemente de um atendimento ao cliente ágil e eficiente, especialmente em processos relacionados a reservas, esclarecimento de dúvidas, cobranças e outras demandas operacionais. Durante a vivência profissional nesse ambiente, observou-se uma significativa ineficiência nos fluxos de atendimento, marcada por tarefas repetitivas que consomem tempo da equipe e reduzem sua produtividade. Esse cenário evidenciou a necessidade de soluções tecnológicas capazes de otimizar o relacionamento com o cliente e automatizar atividades rotineiras, permitindo que os colaboradores se concentrem em demandas mais estratégicas.

Diante desse problema, iniciou-se uma investigação sobre o uso de chatbots para WhatsApp como possível ferramenta de automação. Entretanto, a literatura e pesquisas de mercado indicam que muitos desses sistemas apresentam limitações relevantes, como experiências negativas relatadas pelos usuários, rigidez nos fluxos de atendimento e

dificuldades de interpretação por parte dos bots tradicionais. Além disso, mesmo soluções mais avançadas, baseadas em módulos com Inteligência Artificial, frequentemente exigem alto grau de conhecimento técnico por parte das empresas contratantes, resultando em barreiras à implementação, integração com sistemas internos ou adaptação a websites.

Com base nesse contexto, desenvolveu-se um projeto de chatbot modular com integração personalizável, especialmente voltado ao setor hoteleiro e apoiado por uma equipe técnica responsável por sua implementação completa. A arquitetura proposta utiliza recursos de Inteligência Artificial para oferecer um atendimento mais natural, dinâmico e adaptável às necessidades dos usuários. Para o funcionamento adequado desse sistema, tornou-se indispensável a criação de um banco de dados robusto, capaz de armazenar informações essenciais como cadastro de clientes, embeddings para leitura de IA, histórico de conversas e indicadores de desempenho.

Assim, o objetivo deste trabalho é apresentar o desenvolvimento desse banco de dados, descrevendo sua estrutura, seus propósitos e sua relação com a arquitetura geral do chatbot. A justificativa fundamenta-se na necessidade crescente de automação eficiente e acessível no setor hoteleiro, bem como na superação das limitações identificadas nas soluções existentes. Do ponto de vista metodológico, o estudo apoia-se em análise de mercado, levantamento de requisitos reais de operação, e aplicação de conceitos de engenharia de contexto para prompts, além de fundamentos de arquiteturas lógicas de sistemas inteligentes. O aporte teórico utilizado baseia-se em pesquisas sobre experiência do usuário com chatbots, padrões de sistemas conversacionais inteligentes e práticas de modelagem de dados aplicadas a sistemas de IA.

Dessa forma, este trabalho insere-se no cenário atual de transformação digital na hotelaria, apresentando uma solução tecnológica projetada para atender demandas reais do setor e contribuir para a evolução dos sistemas de atendimento automatizado.

METODOLOGIA

A elaboração do banco de dados proposto neste trabalho teve início com a identificação e análise dos requisitos necessários para sustentar o funcionamento do chatbot modular integrado a um sistema de atendimento via WhatsApp. Como o cadastro completo e detalhado dos clientes permanece nos sistemas internos das empresas contratantes, o escopo deste projeto concentrou-se na criação de uma base de dados enxuta, porém eficiente, destinada ao armazenamento de informações essenciais ao funcionamento do robô. Entre essas informações destacam-se: dados básicos de identificação de clientes, registros utilizados para alimentar o dashboard gerencial, estruturas para suporte ao mecanismo de aprendizagem da Inteligência Artificial e o armazenamento do histórico de conversas.

Para o desenvolvimento do modelo de dados, foi utilizada a plataforma **Supabase**, escolhida por sua robustez, facilidade de integração com aplicações modernas e suporte nativo a banco de dados PostgreSQL. A modelagem foi estruturada por meio da notação **Crow's Foot** (conhecida como "pé de galinha"), que permite representar de forma clara e

intuitiva os relacionamentos entre entidades. Essa abordagem facilitou a definição das tabelas, atributos e cardinalidades, assegurando ao projeto uma organização lógica e alinhada às práticas recomendadas em modelagem conceitual.

O levantamento dos requisitos foi realizado mediante um processo de análise **bottom-up**, partindo da observação detalhada do fluxo de conversa desenvolvido para o chatbot. A partir dessa análise, foi possível identificar as necessidades operacionais mínimas para que o sistema pudesse executar operações CRUD (Create, Read, Update e Delete) de forma segura e eficiente. Esse procedimento garantiu que o banco de dados fosse estruturado de maneira compatível com as demandas reais do atendimento, permitindo a manipulação de registros mediante chaves primárias e proporcionando à IA acesso controlado aos dados para consultas e atualizações.

Essa metodologia também contribuiu para o desenvolvimento de um chatbot menos engessado e mais modular, capaz de interpretar e responder a diferentes cenários de maneira flexível. Essa característica será detalhada nos resultados finais, especialmente no que diz respeito à utilização de embeddings para auxiliar na compreensão contextual do agente de IA.

RESULTADOS FINAIS



No modelo relacional acima está todo o modelo relacional, a primeira vista parece bem simples e pouco profissional porém já se encontra em operação profissional sem apresentar problemas nesta primeira versão.

Este banco de dados de aparência levemente peculiar esconde uma maneira diferente de análise e leitura de maneira muito mais performática que simples querys, neste banco de dados a maior parte da leitura das informações é realizada por coordenadas do tipo embedding. Embedding é a “linguagem de IA”, redes neurais gerais não tem capacidade direta de entender palavras em si, mas sim elas conseguem entender coordenadas que andam perto uma da outra (0.987623, 0987613 ,0987689), estas coordenadas “significam” palavras na leitura delas para geração de mensagens, por exemplo a frase:

“cachorro correndo”, estas palavras possuem forte nexo uma com a outra logo a IA consegue formular esta frase ocorrendo por pesquisas de coordenadas por uma equação matemática de Similaridade de Cossenos :

$$\text{cosine_similarity} = \frac{\vec{A} \cdot \vec{B}}{\|A\| \|B\|}$$

Sendo a primeira linha a similaridade dos vetores, e a segunda linha o tamanho dos vetores e mediante isto a IA irá conseguir ler escrita comum de maneira mais eficaz. Neste trabalho estou utilizando esta ideia para um agente no fluxo denominado agente RAG. Agentes RAG são conceitos derivados de técnicas de engenharia de contexto na montagem de agentes, sua função é ter acesso a estes embeddings para leitura de documentos inteiros em texto para gerar “dicas de resposta” para a estrutura de agentes que irá gerar um procedimento ou retornar um valor, por exemplo, digamos que você tenha um hotel, este hotel existem políticas como: proibido fumar, proibido pets, uma vaga de estacionamento por pessoa e não suporte a deficientes físicos, quando colocado estas informações em um documento docs ligado ao sistema de agentes e atualizado, a tabela documents apaga a coluna referente ao id do hotel e cria uma nova linha com o mesmo id, atualizando essas informações no banco de dados com esta coluna em notação embedding, traduzida por uma parte do sistema.

Logo quando uma mensagem é recebida via whatsapp o conteúdo da mensagem passa por uma parte do fluxo do agente RAG para interpretação da mensagem, análise do documento para encontrar correlação entre as partes e enviar para a estrutura de agentes que irá gerar a resposta posteriormente. Na prática utilizamos um documento .doc para colocar os scripts de mensagem que nosso cliente utiliza para o bot poder basear sua resposta e também políticas internas do estabelecimento, assim podendo utilizar a mesma arquitetura para vários negócios com regras semelhantes mudando apenas a base de dados que alimenta todo este fluxo de conversa. Para realizar o cadastro de clientes utilizei a tabela Clientes com informações normais como created_at, id, whatsapp, nome(nome fixado no whatsapp), nome_recebido(nome passado pelo cliente durante a conversa, ultimo_acesso, email, CPF, paused(indicador booleano para controle de pause da conversa automatizada por chat), e indicador_de_reserva(indicador para saber se o cliente ja está hospedado ou não). A tabela n8n_chathistories utiliza uma chave estrangeira com a tabela de clientes aonde ele armazena o contexto das últimas mensagens(se existentes) para continuar a conversa. As demais tabelas de são apenas para controle de dashboard, são colunas para o sistema conseguir armazenar, buscar e atualizar as informações para o dashboard, estas tabelas possuem dados gerais, ou seja possuem dados de massas de usuários, uma estrutura de agentes de IA que tem acesso a todas as mensagens, consultas de API, e contexto de clientes, alimentam estas tabelas para que o dashboard tenha uma visão ampla de todas as conversas de uma maneira metrificada para o cliente poder realizar uma análise geral de seu negócio e manejear estratégias para melhora em ponto específicos.

CONCLUSÃO

Em suma este projeto desenvolveu um banco de dados completo voltado para a integração de uma estrutura de agentes de inteligência artificial, com suas funcionalidades de armazenamento em vetores para informações massivas, umas tabela de clientes alimentado pelo modelo e uma tabela de histórico de contexto das conversas, além das tabelas voltadas para manuseio do dashboard e de seus gráficos que serão controlados

por estruturas de agentes que irá analisar as massas de conversas para ajudar ao contratante realizar melhorias na sua operação com base nestes dados.

REFERÊNCIAS BIBLIOGRÁFICAS

MINDS AND MACHINE <https://link.springer.com/article/10.1007/s11023-020-09537-4>

Embedding in iot <https://link.springer.com/article/10.1007/s00607-018-0680-z>