



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus “José Santilli Sobrinho”**

JOÃO FERNANDO CORREIA DE JESUS

TECNOLOGIA DE CHATBOT PARA ATENDIMENTO A CLIENTES

**Assis/SP
2024**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus “José Santilli Sobrinho”**

JOÃO FERNANDO CORREIA DE JESUS

TECNOLOGIA DE CHATBOT PARA ATENDIMENTO A CLIENTES

Projeto de pesquisa apresentado ao Curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e à Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): João Fernando Correia de Jesus
Orientador(a): Me. Guilherme de Cleve Farto

**Assis/SP
2024**

Jesus, João Fernando Correia de

J585t Tecnologia de chatbot para atendimento a clientes / João Fernando Correia de Jesus. --Assis, 2024.

38p.

Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2024.

Orientador: Prof. Me. Guilherme de Cleve Farto.

1. Automação. 2. Serviço ao cliente. 3. Aplicações de computador. I Farto, Guilherme de Cleve. II Título.

CDD 001.64

AGRADECIMENTOS

Primeiramente, agradeço a Deus por me conceder a oportunidade de estar aqui, por me dar força, foco e determinação para chegar até este momento. Agradeço, também, à minha mãe, Adriana, à minha avó, e ao meu pai, Fábio, por todo o apoio incondicional e por acreditarem em mim ao longo dessa jornada. Sem eles, eu não estaria aqui.

Gostaria de expressar minha gratidão ao meu orientador e amigo, Guilherme de Cleve Farto, pela paciência, pelo constante apoio e pela orientação essencial ao longo deste projeto. Seu comprometimento e conhecimento foram fundamentais para a realização deste trabalho.

Aos meus colegas de sala e companheiros de turma, agradeço pela camaradagem, pela troca de conhecimentos e pela descontração durante toda a trajetória acadêmica. O network e as amizades que construímos foram parte crucial dessa experiência enriquecedora.

RESUMO

O desenvolvimento de um chatbot integrado com o WhatsApp para um restaurante visa aprimorar o atendimento ao cliente, oferecendo uma experiência mais eficiente e personalizada. Este trabalho detalha a construção de um sistema que permite aos clientes realizar pedidos, consultar históricos e gerenciar informações de forma autônoma e dinâmica. A implementação do chatbot, utilizando tecnologias como Node.js, Express.js, e PostgreSQL, traz benefícios significativos para o restaurante, incluindo a automação de processos, redução de erros, aumento da produtividade e uma melhor experiência do usuário, contribuindo para a fidelização dos clientes e a otimização da gestão do restaurante.

Palavras-chave: *ChatBot*, Atendimento ao Cliente, Restaurante, Automação, Node.js;

ABSTRACT

The development of a chatbot integrated with WhatsApp for a restaurant aims to improve customer service by offering a more efficient and personalized experience. This work details the construction of a system that allows customers to place orders, consult histories and manage information autonomously and dynamically. The implementation of the chatbot, using technologies such as Node.js, Express.js, and PostgreSQL, brings significant benefits to the restaurant, including process automation, error reduction, increased productivity and a better user experience, contributing to customer loyalty and the optimization of restaurant management.

Palavras-chave: *ChatBot*, Customer Service, Restaurant, Automation, Node.js;

LISTA DE ILUSTRAÇÕES

Figura 1: Diagrama de Caso De Uso – Cliente.....	17
Figura 2: Diagrama de Caso De Uso – <i>ChatBot</i>	18
Figura 3: Diagrama de Caso De Uso – Administrador.....	19
Figura 4: Diagrama Geral de Casos de Uso.....	20
Figura 5: UC Fazer Pedido.....	21
Figura 6: UC Consulta Histórico de Pedidos.....	22
Figura 7: UC Manter Cliente.....	23
Figura 8: UC Atualizar Cardápio.....	24
Figura 9: Diagrama de Classes	25
Figura 10: Diagrama Entidade Relacionamento.....	26
Figura 11: Configuração do Servidor Express.....	27
Figura 12: Criação e Gerenciamento de Clientes.....	28
Figura 13: Configuração Inicial e Geração de QR Code.....	29
Figura 14: Manipulação de Mensagens e Fluxo de Conversa.....	30
Figura 15: Lógica para Processar Pedidos.....	31
Figura 16: Teste Usuário Cadastrado Visualizando Pedidos Anteriores.....	32
Figura 17: Teste Usuário Cadastrado Visualizando Pedidos Anteriores.....	34

SUMÁRIO

1. INTRODUÇÃO.....	8
1.1 OBJETIVOS.....	9
1.1.1 OBJETIVO GERAL.....	9
1.1.2 OBJETIVOS ESPECÍFICOS.....	9
1.2 PÚBLICO ALVO.....	10
1.3 JUSTIFICATIVAS.....	10
1.4 ESTRUTURA DO TRABALHO.....	11
2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO.....	12
2.1 TECNOLOGIAS.....	13
2.1.1 UML.....	13
2.1.2 Diagrama de Casos de Uso.....	13
2.1.3 Diagrama de Classe.....	13
2.1.4 Astah Community.....	14
2.1.5 Node.js.....	14
2.1.6 Framework Express.js.....	14
2.1.7 Banco de Dados PostgreSQL.....	15
3. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA.....	16
3.1 Lista de Requisitos.....	16
3.2 DIAGRAMA DE CASOS DE USO.....	20
3.2.1 UC 01 Fazer Pedido.....	21
3.2.2 UC 02 Consulta Histórico de Pedidos.....	22
3.2.3 UC 03 Manter Cliente.....	23
3.2.4 UC 04 Atualizar Cardápio.....	24
3.3 DIAGRAMA DE CLASSE.....	25
3.4 DIAGRAMA ENTIDADE RELACIONAMENTO.....	26
4. DESENVOLVIMENTO DO TRABALHO.....	27
4.1 Estrutura do Back-End.....	27
4.1.1 Configuração do Servidor Express.js.....	27
4.1.2 Criação e Gerenciamento de Clientes.....	28
4.2 Integração com o WhatsApp.....	29
4.2.1 Configuração Inicial e Geração de QR Code.....	29
4.2.2 Manipulação de Mensagens e Fluxo de Conversa.....	30
4.3 Processamento de Pedidos e Pagamentos.....	31
4.3.1 Lógica para Processar Pedidos.....	31
4.4 ChatBot em funcionamento.....	32
4.4.1 Usuário Cadastrado Visualizando Pedidos Anteriores.....	32
4.4.2 Usuário Fazendo um Pedido e Confirmando o Pagamento.....	34
5. CONCLUSÕES.....	36
5.1 TRABALHOS FUTUROS.....	36
REFERÊNCIAS.....	38

1. INTRODUÇÃO

No cenário contemporâneo, há uma era digital profundamente interconectada, onde a tecnologia desempenha um papel preponderante em nossas interações cotidianas. Este fenômeno é especialmente pronunciado na indústria de restaurantes, onde a satisfação do cliente e a qualidade do serviço são cruciais para o sucesso estabelecido de um estabelecimento. Nesse contexto, a exploração de soluções inovadoras capazes de aprimorar e otimizar os processos de atendimento ao cliente se revela imperativa.

Destacando-se como uma ferramenta promissora para redefinir a experiência na indústria de restaurantes, os *ChatBot* emergem como assistentes virtuais capazes de simular conversas humanas, proporcionando respostas automáticas por meio de mensagens de texto. Sua destreza em compreender a linguagem natural possibilita a oferta de um serviço personalizado e responsivo, disponível ininterruptamente, 24 horas por dia.

Segundo Adaniya e Aquino (2018), “ Um *ChatBot* é um sistema de conversação por computador que interage com usuários humanos por meio de uma linguagem conversacional natural. Os primeiros chatterbots eram aplicados estritamente a estudos acadêmicos. Atualmente, eles são considerados alternativas capazes de desempenhar o papel de facilitadores em diversas aplicações como, por exemplo, uso pedagógico, comercial, social, ensino a distância entre outros ”.

Particular atenção será dedicada aos casos de sucesso de restaurantes que já incorporaram chatbots em suas operações de atendimento ao cliente. Por meio dessa análise de casos, pretendemos destacar as melhores práticas adotadas e extrair lições valiosas que possam orientar gestores e profissionais na busca pela melhoria contínua da experiência do cliente e, por conseguinte, no impulsionamento de negócios bem-sucedidos.

Contudo, a implementação ética de *ChatBots* não pode ser negligenciada. Questões relativas à transparência e privacidade dos dados dos clientes serão cuidadosamente discutidas para assegurar que a introdução dessas inovações ocorra de maneira responsável e sustentável.

A relevância dos *ChatBots* transcende setores, demonstrando promessas em áreas como atendimento bancário e suporte ao cliente online. No entanto, é notável a escassez de pesquisas que explorem plenamente seu potencial na indústria de serviços alimentícios. Ao preencher essa lacuna, este trabalho acadêmico visa fornecer uma visão aprofundada sobre como os *ChatBots* podem ser efetivamente incorporados para otimizar a experiência do cliente em restaurantes.

Como afirmam Ghidini e Mattos (2018), "as respostas do robô inteligente fazem parecer que há uma pessoa do outro lado da tela conversando normalmente com o usuário, tanto é verdade, que em alguns casos, este robô recebe um nome e uma foto no chat de atendimento". Além disso, conforme Lugli e Filho (2020) destacam, os chatbots desempenham um papel crucial ao evidenciar que o uso dessa tecnologia é vantajoso para empresas que buscam a excelência no atendimento ao cliente, aliviando a sobrecarga de problemas levados aos operadores.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O objetivo geral deste trabalho foi desenvolver e avaliar um *ChatBot* de atendimento ao cliente, especificamente projetado para a indústria de restaurantes e serviços alimentícios, com o propósito de aprimorar a eficiência operacional, personalização dos serviços e automação de serviços.

1.1.2 OBJETIVOS ESPECÍFICOS

Este projeto tem como objetivo principal a criação e implementação de um *ChatBot* personalizado para o setor de atendimento ao cliente no contexto do restaurante. A iniciativa busca empregar técnicas de automação e integração com sistemas de banco de dados para proporcionar interações eficientes, abrangendo áreas como pedidos, cadastro de clientes e informações sobre o status dos pedidos realizados.

1.2 PÚBLICO ALVO

O *ChatBot* de atendimento ao cliente tem como foco principal os usuários que frequentam o restaurante, em especial aqueles que realizam pedidos e interagem com os serviços oferecidos. Além disso, destina-se aos proprietários do estabelecimento, visando simplificar a gestão operacional, promover uma comunicação mais eficiente com os clientes e otimizar as operações diárias do restaurante.

1.3 JUSTIFICATIVAS

No cenário atual, a tecnologia desempenha um papel central no atendimento ao cliente, com a pandemia global tendo intensificado significativamente a necessidade de soluções eficazes para interações remotas. A implementação de um chatbot de atendimento ao cliente é, portanto, altamente relevante e oportuna, especialmente no setor de restaurantes. Este setor, que se viu fortemente impactado por restrições de distanciamento social, busca constantemente por meios de manter e melhorar o atendimento ao cliente sem comprometer a segurança e a qualidade do serviço.

Um *chatbot* oferece uma solução prática e eficiente, automatizando processos como pedidos e o fornecimento de informações sobre o cardápio, reduzindo a carga de trabalho dos funcionários e melhorando a experiência do cliente. Além disso, a adoção de um chatbot permite que restaurantes operem de maneira mais flexível e responsiva, atendendo a um volume maior de interações com rapidez e precisão, e ajudando a manter a fidelidade dos clientes em tempos de mudança e incerteza.

1.4 ESTRUTURA DO TRABALHO

A estrutura deste trabalho é composta das seguintes partes:

- **Capítulo 1 - Introdução:** Este capítulo estabelece o contexto da área de estudo, os objetivos e justificativas.
- **Capítulo 2 - Tecnologias e Ferramentas de Desenvolvimento:** Neste capítulo, são abordadas as tecnologias e recursos utilizados no desenvolvimento.
- **Capítulo 3 - Análise e Especificação do Sistema:** Este capítulo é sobre a análise realizada e apresenta as especificações do sistema de forma detalhada.
- **Capítulo 4 - Desenvolvimento do Sistema:** Neste capítulo, são apresentados os detalhes do desenvolvimento do sistema.
- **Capítulo 5 - Conclusão:** Fornece uma explicação dos principais pontos abordados no desenvolvimento deste trabalho, destacando os resultados alcançados e sugerindo direções para trabalhos futuros.

2. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

A metodologia adotada para o desenvolvimento do *ChatBot* de atendimento ao cliente personalizado compreende cinco etapas cruciais, que foram fundamentais para a construção e integração eficaz da solução:

Análise de Requisitos e Documentação: Inicialmente, foi realizada uma análise aprofundada dos requisitos do projeto, considerando as funcionalidades essenciais para a aplicação do *ChatBot*. Essa etapa incluiu uma revisão detalhada da documentação disponível, garantindo um entendimento claro do escopo do projeto e das necessidades dos usuários finais

Modelagem do Sistema: Na segunda etapa, foram criados artefatos UML, como o Diagrama de Caso de Uso, Diagrama de Classes e Diagrama de Sequência. Esses diagramas foram fundamentais para visualizar e modelar a arquitetura do sistema, ajudando a definir as interações entre o *ChatBot* e os componentes do sistema.

Desenvolvimento Back-End com Node.js e Express: A terceira etapa envolveu a criação da estrutura do back-end. Utilizando Node.js e o framework Express, foram desenvolvidos os serviços necessários para a integração do ChatBot, gerenciamento de sessões de usuário, e manipulação de dados de pedidos e clientes. A escolha dessas tecnologias foi baseada na facilidade de integração com outras ferramentas.

Integração com a API do WhatsApp e Banco de Dados PostgreSQL: Uma fase crucial foi a integração com a API do WhatsApp, utilizando a biblioteca `whatsapp-web.js`, para permitir as interações diretas com os clientes. Além disso, foi configurada uma base de dados PostgreSQL para armazenar informações de clientes e pedidos.

Testes e Validação: Finalmente, a última etapa compreendeu a realização de testes, essenciais para garantir que o *ChatBot* funcionasse conforme o planejado, respeitando as regras de negócios definidas e assegurando um desempenho sólido e confiável. Os testes permitiram identificar e corrigir eventuais falhas, melhorando a robustez e a experiência geral do usuário.

2.1 TECNOLOGIAS

2.1.1 UML

Segundo Fowler (2011, p. 25), a UML (Unified Modeling Language) é um conjunto de anotações gráficas estabelecido por um modelo substituto único, que auxilia na definição e no projeto de sistemas de software, especialmente aqueles construídos utilizando o estilo orientado a objetos. A UML possui diferentes significados, adequados à sua própria história e às diversas abordagens de um processo de engenharia de software eficiente.

2.1.2 Diagrama de Casos de Uso

De acordo com Fowler (2011), um diagrama de casos de uso é uma representação simples do conjunto de casos de uso de um sistema. Ele é similar ao diagrama de contexto utilizado nos métodos estruturados, pois mostra o limite do sistema e suas interações com o mundo exterior. O diagrama de casos de uso ilustra os atores, os casos de uso e seus relacionamentos.

2.1.3 Diagrama de Classe

Ainda segundo Fowler (2011), os diagramas de classes representam os tipos de objetos existentes no sistema e as diversas conexões estáticas entre eles. Além disso, os diagramas de classe demonstram as propriedades, métodos de uma classe, e as restrições relacionadas à forma como os objetos são manipulados.

2.1.4 Astah Community

Para auxiliar na criação de diagramas UML, foi utilizada a ferramenta Astah Community. Segundo Lima (2016), o Astah Community é um software de modelagem UML (Unified Modeling Language – Linguagem de Modelagem Unificada) com suporte a UML 2, desenvolvido pela Change Vision, Inc., disponível para sistemas operacionais Windows 64 bits. Originalmente conhecido como JUDE, um acrônimo para Java and UML Developers Environment (Ambiente para Desenvolvedores UML e Java).

2.1.5 Node.js

Segundo a Alura (2023), o JavaScript é uma das linguagens mais utilizadas atualmente, sendo a base para dezenas de frameworks populares na comunidade de desenvolvimento. Conhecido principalmente por sua aplicação no Front-end (*client side*) para criar aplicações web mais interativas, o JavaScript evoluiu ao ponto de ser utilizado também no Back-end. Esse avanço nas tecnologias web possibilitou o surgimento do Node.js, que é um ambiente de execução para código JavaScript no lado do servidor (server side). Na prática, isso significa que o Node.js permite a criação de aplicações standalone (autossuficientes) em uma máquina servidora, sem a necessidade de um navegador.

2.1.6 Framework Express.js

De acordo com o Blog XP Educação (2023), o Express JS é um framework que utiliza a linguagem de programação JavaScript para desenvolver aplicações baseadas em Node JS. Esta ferramenta é amplamente adotada pelos desenvolvedores devido à sua capacidade de agilizar o fluxo de trabalho e permitir a criação de servidores web de maneira mais estruturada e limpa. A popularidade do Express JS é tamanha que empresas como Uber e IBM o utilizam no desenvolvimento de seus aplicativos.

2.1.7 Banco de Dados PostgreSQL

De acordo com a Rock Content (2023), o PostgreSQL é uma ferramenta que atua como sistema de gerenciamento de bancos de dados relacionais, focando na implementação da linguagem SQL em estruturas e garantindo a conformidade com padrões de ordenação de dados. Nos últimos anos, seu uso cresceu consideravelmente devido à sua praticidade e alta compatibilidade com diferentes padrões de linguagem.

O PostgreSQL destaca-se pela adequação a padrões de conformidade, facilitando a construção de bancos de dados otimizados, além de armazenar informações de forma segura e permitir a restauração delas conforme solicitado por outras aplicações integradas.

Ainda segundo a Rock Content (2023), o PostgreSQL é eficaz no gerenciamento de grandes volumes de solicitações e cargas de trabalho intensas, sendo ideal para sites de alto tráfego, como e-commerces. O sistema organiza e gerencia os dados de maneira eficiente, permitindo que usuários executem consultas de forma simples, sem necessidade de acessar diretamente o banco de dados. Dessa forma, o PostgreSQL atua como um verdadeiro organizador de informações, proporcionando um acesso rápido e seguro para consultas e configurações.

2.1.8 WhatsApp Business API

O WhatsApp, com mais de 2 bilhões de usuários ativos mensais, se destaca como a plataforma ideal para a integração de chatbots em negócios. Por meio da API do WhatsApp Business, o chatbot é capaz de enviar e receber mensagens de forma segura e eficiente, ampliando as possibilidades de atendimento ao cliente e promovendo o crescimento do negócio. O chatbot funciona como um assistente virtual para o restaurante, oferecendo atendimento contínuo, respondendo a perguntas frequentes, fornecendo informações sobre o cardápio, horários de funcionamento, promoções e eventos, liberando a equipe para se concentrar em tarefas mais complexas.

3. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

3.1 Lista de Requisitos

- Manter Informações sobre o Cardápio;
- Realizar Pedidos;
- Manter Histórico de Pedidos;
- Manter Cliente;

Para melhor visualização dos requisitos da aplicação foram elaborados alguns diagramas, apresentados na Figura 1, Figura 2 e Figura 3.

Como elemento primário da UML, foi utiliza-se o Diagrama de Casos de Uso para especificar o comportamento de um sistema ou de parte dele e é uma descrição de um conjunto de sequências de ações (BOOCH; JACOBSON; RUMBAUGH, 2005). Na Figura 1, Figura 2 e Figura 3 são apresentados os Diagramas de Casos de Uso que exibem algumas funcionalidades/recursos do projeto proposto.

A Figura 1 ilustra um Diagrama de Casos de Uso relacionado ao "Cliente", detalhando as interações principais que este pode realizar dentro do sistema, como a realização de pedidos, pagamentos, consulta de histórico de pedidos e cancelamento de pedidos.

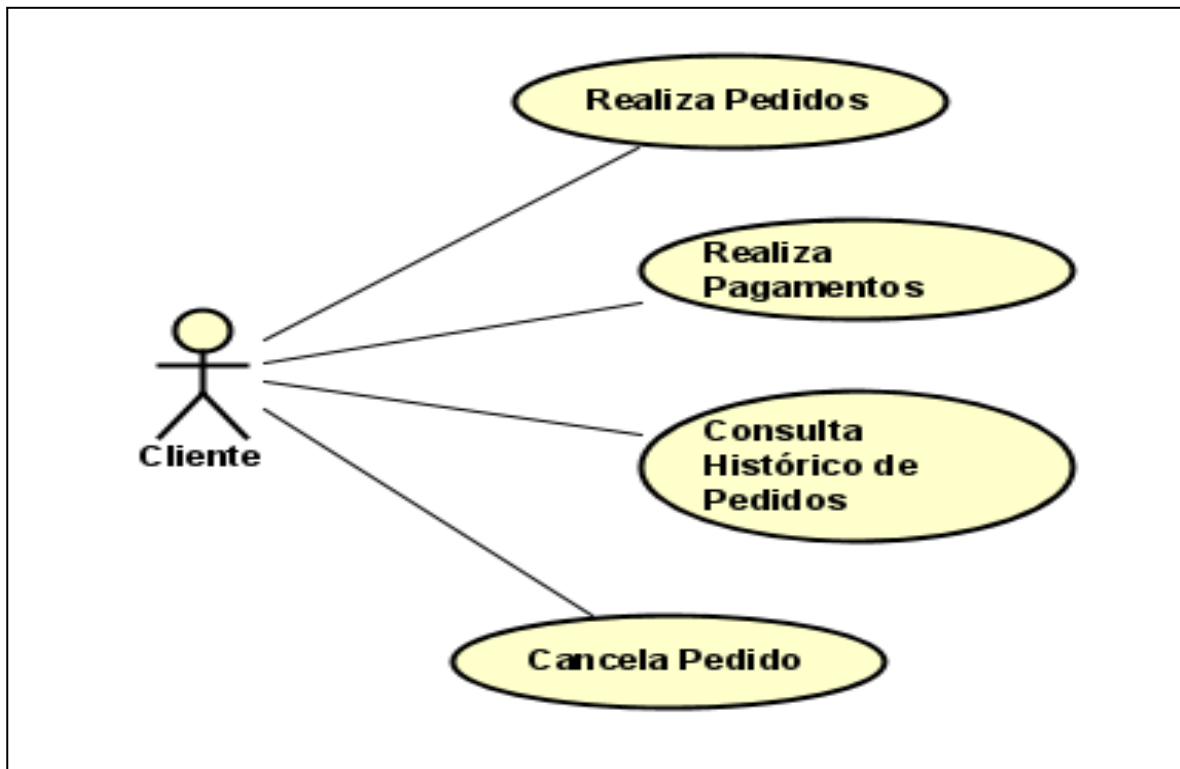


Figura 1 – Diagrama de Caso De Uso – Cliente

A Figura 2 demonstra um Diagrama de Casos de Uso para o "Chatbot", exibindo as principais funcionalidades que o sistema automatizado pode executar, como a consulta do menu, manutenção de informações do cliente, cancelamento de pedidos e emissão de pedidos.

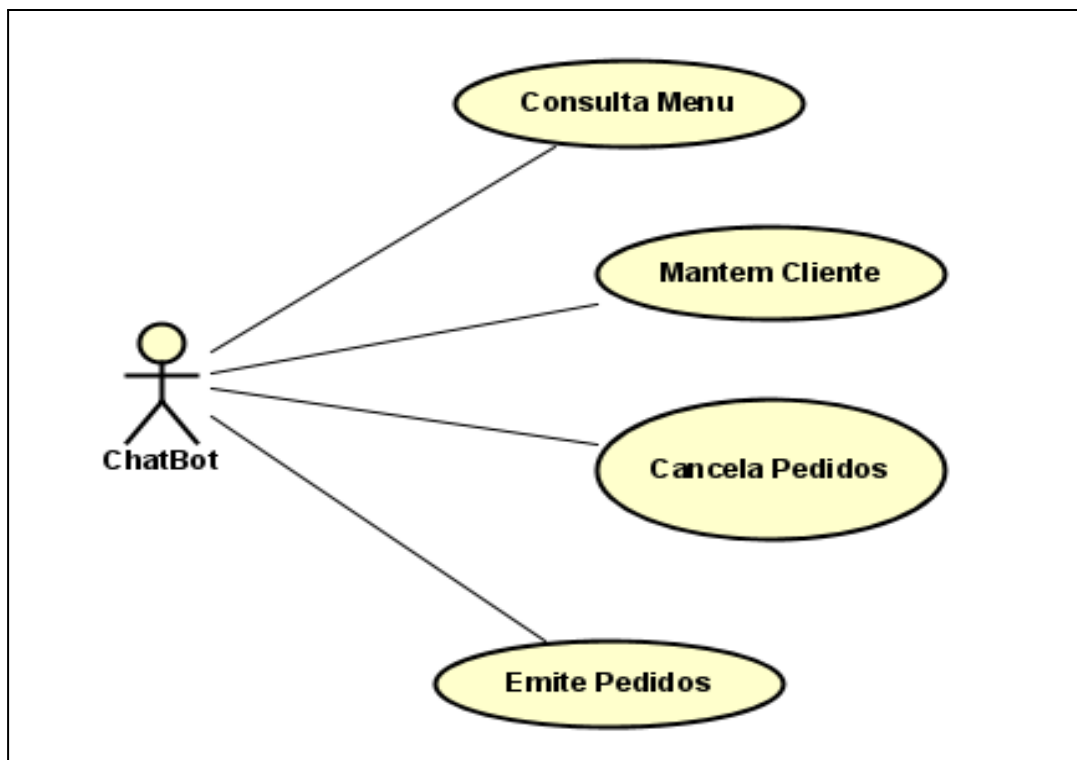


Figura 2 – Diagrama de Caso De Uso – *ChatBot*

A Figura 3 ilustra um Diagrama de Casos de Uso para o "Administrador", destacando as funções administrativas essenciais, como a gerência de preços, a atualização das informações do cliente, a administração de pedidos e a manutenção do cardápio do restaurante.

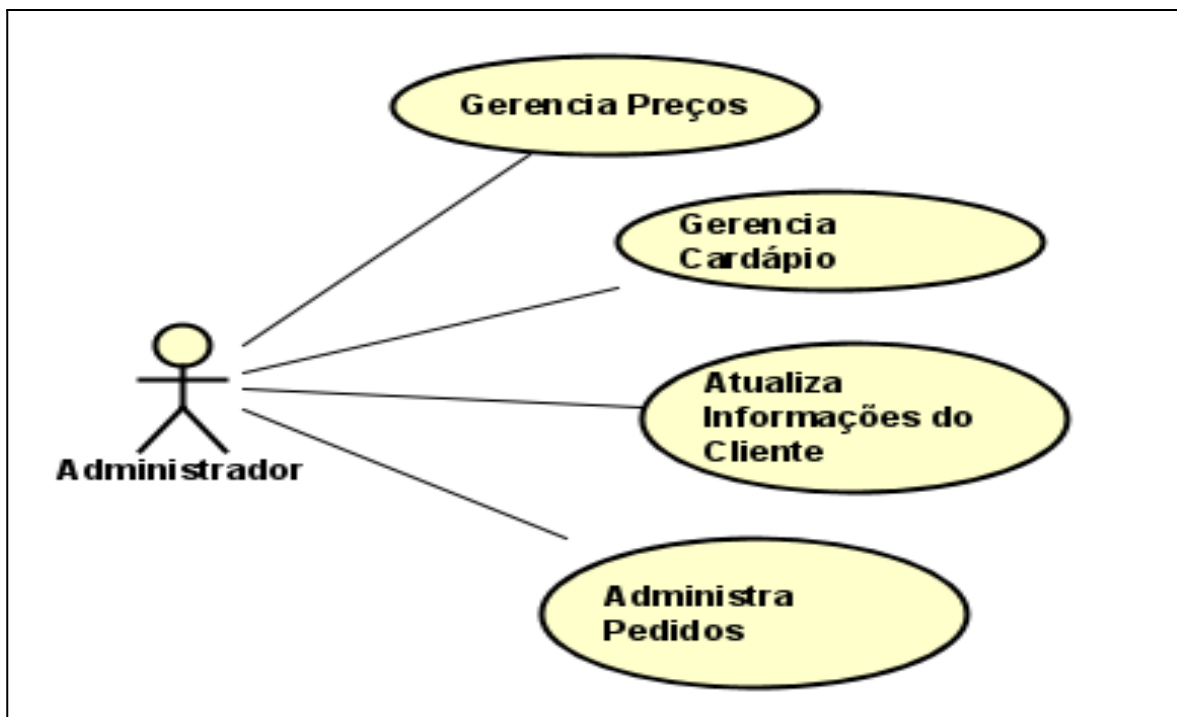


Figura 3 – Diagrama de Caso De Uso – Administrador

3.2 DIAGRAMA DE CASOS DE USO

Segundo Fowler (2011), um diagrama de caso de uso é como um simples gráfico do conjunto de casos de uso. O diagrama de casos de uso exhibe os atores, os casos de uso e os relacionamentos.

A Figura 4 apresenta o Diagrama de Casos de Uso geral do Sistema

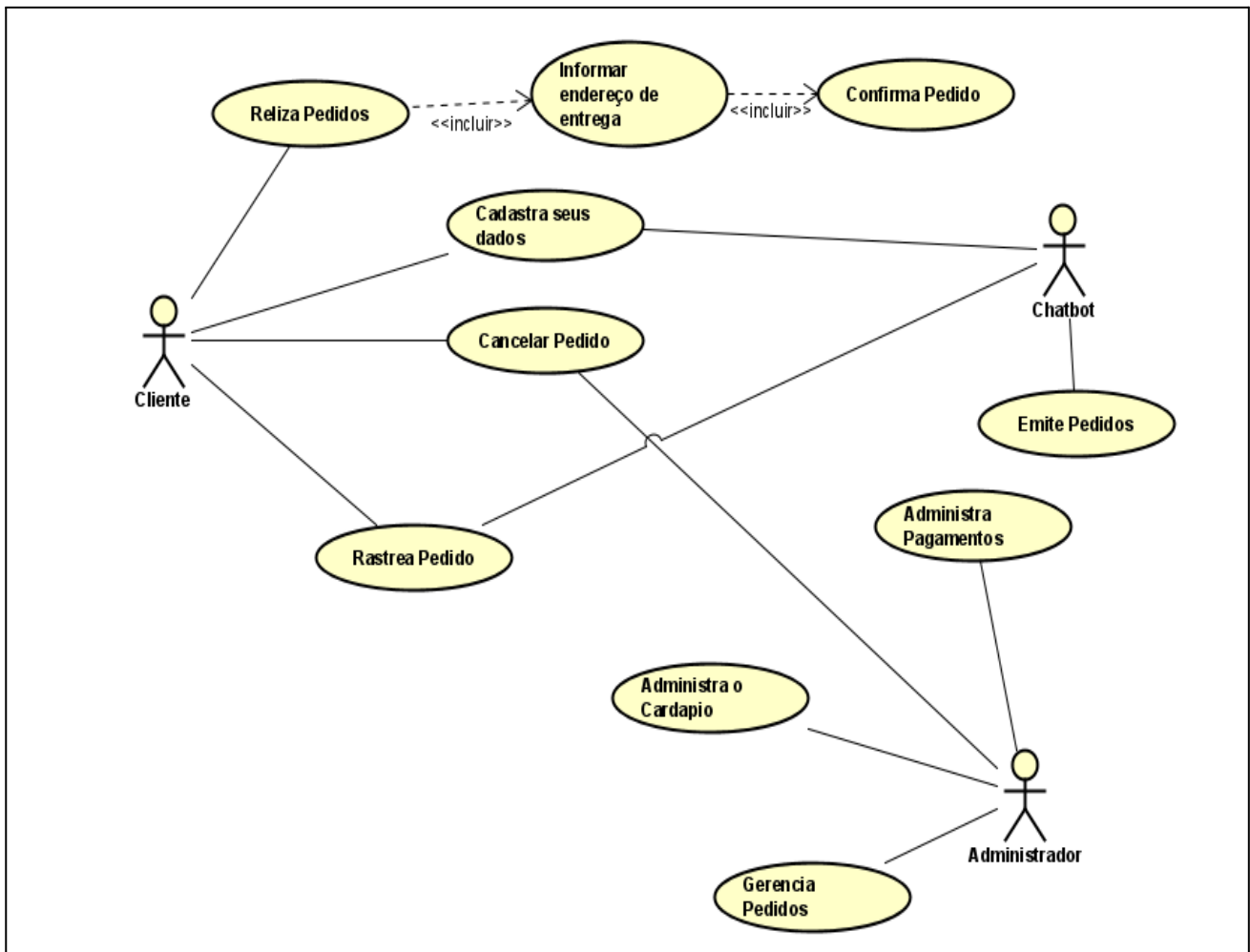


Figura 4 - Diagrama Geral de Casos de Uso

3.2.1 UC 01 Fazer Pedido



Figura 5 - UC Fazer Pedido

Finalidade/Objetivo	Fazer Pedido
Atores	Cliente <i>ChatBot</i>
Pré-condições	-cliente deve ter acesso à internet. -cliente deve ter um dispositivo compatível com o chatbot. -restaurante deve estar com o sistema de pedidos online ativado.
Evento inicial	O cliente inicia o <i>chatbot</i> e seleciona o restaurante desejado.
Fluxo Principal	<ul style="list-style-type: none"> ● O <i>chatbot</i> apresenta o cardápio do restaurante ao cliente. ● O cliente escolhe os itens do cardápio e informa as quantidades desejadas. ● O <i>chatbot</i> oferece opções de personalização para os itens (se disponível). ● O cliente informa o endereço de entrega. ● O <i>chatbot</i> sugere opções de método de pagamento. ● O cliente seleciona o método de pagamento e confirma o pedido. ● O <i>chatbot</i> processa o pagamento e envia a confirmação do pedido para o cliente e Administrador. ● O <i>chatbot</i> informa o tempo de entrega ao cliente.
Fluxo Alternativo	a)Se o cliente insere uma quantidade que não está disponível ou é inválida (ex.: um número negativo ou zero).O chatbot informa que a quantidade é inválida.O cliente é solicitado a inserir uma quantidade válida novamente.

3.2.2 UC 02 Consulta Histórico de Pedidos



Figura 6 - UC Consulta Histórico de Pedidos

Finalidade/Objetivo	Consulta Histórico de Pedidos
Atores	Cliente <i>ChatBot</i>
Pré-condições	<ul style="list-style-type: none"> - Cliente deve ter acesso à internet. - Cliente deve ter um dispositivo compatível com o chatbot. - Cliente deve ter realizado pedidos anteriores.
Evento inicial	<p>O cliente inicia o chatbot e solicita o histórico de pedidos.</p>
Fluxo Principal	<ul style="list-style-type: none"> ● O <i>chatbot</i> verifica o número de telefone do cliente e consulta o banco de dados. ● O <i>chatbot</i> retorna uma lista de pedidos anteriores realizados pelo cliente, incluindo detalhes como itens pedidos, data e valor total. ● O cliente pode selecionar um pedido específico para ver mais detalhes, se desejar. ● O chatbot apresenta detalhes adicionais do pedido selecionado (se aplicável).
Fluxo Alternativo	<p>a) Se o cliente não tiver pedidos anteriores, o <i>chatbot</i> informa que não há histórico de pedidos disponíveis.</p>

3.2.3 UC 03 Manter Cliente

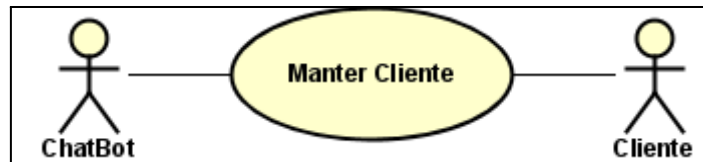


Figura 7 - UC Manter Cliente

Finalidade/Objetivo	Manter Cliente
Atores	Cliente <i>ChatBot</i>
Pré-condições	<ul style="list-style-type: none"> - Cliente deve ter acesso à internet. - Cliente deve ter um dispositivo compatível com o chatbot. - O cliente precisa ser identificado pelo número de telefone ou outro identificador único.
Evento inicial	O cliente inicia o chatbot e solicita a atualização de suas informações cadastrais ou o cadastro de um novo cliente.
Fluxo Principal	<ul style="list-style-type: none"> ● O <i>chatbot</i> solicita as informações necessárias para atualizar ou criar um novo cadastro (nome, telefone). ● O cliente fornece as informações solicitadas. ● O <i>chatbot</i> valida as informações fornecidas (como número de telefone correto ou formato de endereço). ● Se as informações forem válidas, o <i>chatbot</i> atualiza ou cria o cadastro do cliente no banco de dados. ● O chatbot confirma a atualização ou criação do cadastro com o cliente.

3.2.4 UC 04 Atualizar Cardápio

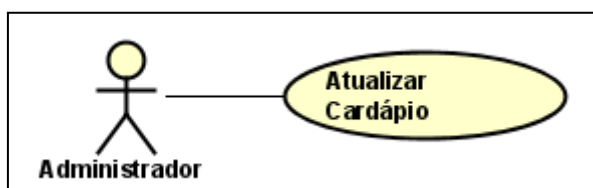


Figura 8 - UC Atualizar Cardápio

Finalidade/Objetivo	Atualizar Cardápio
Atores	Administrador
Pré-condições	- O administrador deve ter acesso ao código fonte do sistema. - O sistema deve estar em desenvolvimento ou manutenção.
Evento inicial	O administrador decide realizar uma atualização no cardápio do restaurante.
Fluxo Principal	<ul style="list-style-type: none"> ● O administrador acessa o código fonte do sistema onde o cardápio está definido. ● O administrador localiza a seção do código que contém os itens do cardápio. ● O administrador edita o código para adicionar um novo item, alterar detalhes de um item existente ou remover um item do cardápio. ● Após fazer as alterações necessárias, o administrador salva o código e realiza testes locais para verificar se as mudanças foram aplicadas corretamente. ● Se os testes estiverem corretos, o administrador faz o commit das mudanças no repositório de código e realiza o deploy do sistema atualizado. ● O sistema é atualizado com o novo cardápio, refletindo as mudanças feitas diretamente no código.
Fluxo Alternativo	<p>a) Se o administrador encontrar um erro ao editar o código, ele pode corrigir o erro e tentar novamente antes de fazer o commit das alterações.</p> <p>b) Se o administrador decidir não prosseguir com as mudanças, ele pode desfazer as alterações no código antes de salvar e sair do editor de código.</p>

3.3 DIAGRAMA DE CLASSE

Os diagramas de classes de acordo com Fowler (2011), os diagramas de classes descrevem os vários tipos de objetos que existem no sistema, bem como os vários tipos de conexões de estado que existem entre eles. Os diagramas de classes também mostram as propriedades e procedimentos de uma classe, bem como as restrições que se aplicam à maneira como os objetos são relacionados. Na imagem a seguir, será exibido o diagrama de classe, que contém os principais atributos e métodos que será necessário em cada objeto para a aplicação.

A Figura 7 apresenta o Diagrama de Classes Geral do sistema

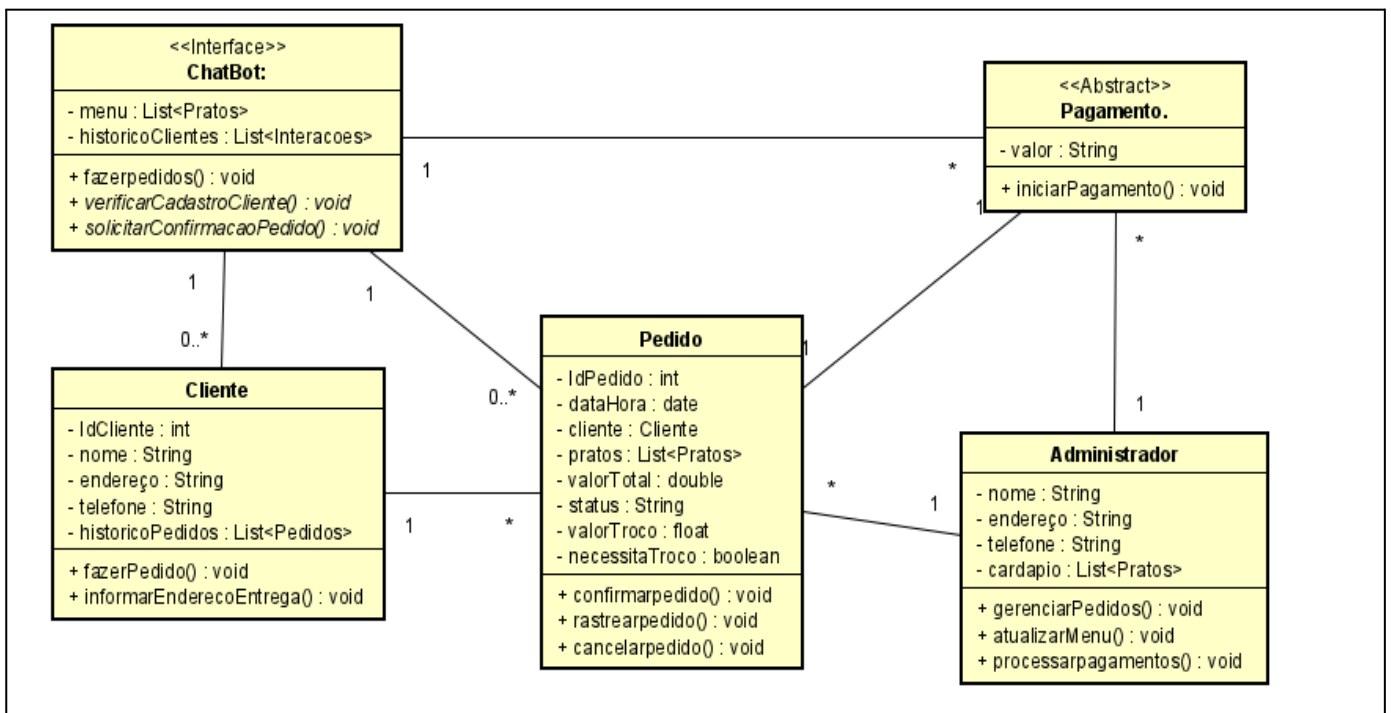


Figura 9 - Diagrama de Classes

3.4 DIAGRAMA ENTIDADE RELACIONAMENTO

O Diagrama de Entidade e Relacionamento é uma ferramenta para projetar ou depurar bancos de dados relacionais. A seguir está um modelo de fluxograma que descreve as entidades usadas para moldar fluxos de processos, processos de negócios ou operações internas que incluem a captura de ações e seus resultados com base em mudanças no estado do objeto. A Figura 8 apresenta o Diagrama de Entidade Relacionamento do Sistema, demonstrando a estrutura de Tabelas e Colunas do Sistema.

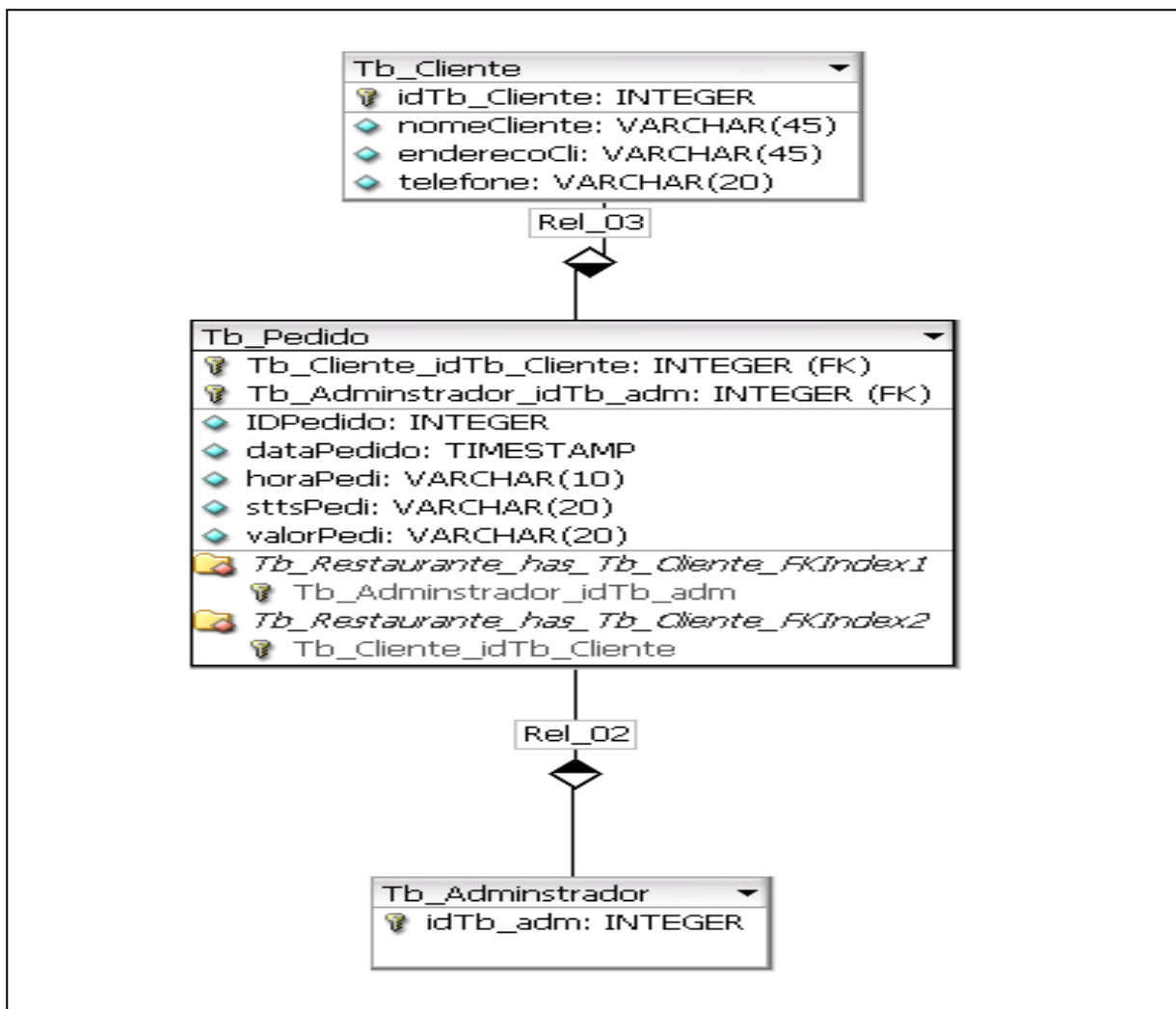


Figura 10 - Diagrama Entidade Relacionamento

4. DESENVOLVIMENTO DO TRABALHO

O desenvolvimento deste trabalho foi estruturado em diferentes etapas, contemplando as tecnologias específicas adotadas para criar um sistema de atendimento ao cliente eficiente por meio de um chatbot. A seguir, detalhamos o processo de construção do *back-end* e a integração com o serviço de mensagens WhatsApp.

4.1 Estrutura do Back-End

4.1.1 Configuração do Servidor Express.js

O servidor foi configurado utilizando o *framework Express.js* para criar uma *API RESTful* que se comunica com o banco de dados *PostgreSQL*. Essa *API* é responsável por gerenciar as rotas de clientes e pedidos.

```
const express = require('express');
const pool = require('./db'); // Arquivo de configuração do banco de dados

const app = express();
app.use(express.json()); |
```

Figura 11 - Configuração do Servidor Express

A função `express()` cria uma nova aplicação Express. O *middleware* `express.json()` é utilizado para processar os corpos de requisição com formato *JSON*. Esta configuração é essencial para a comunicação entre o *front-end* (*chatbot*) e o *back-end* (servidor).

4.1.2 Criação e Gerenciamento de Clientes

Neste módulo, o código gerencia a criação de novos clientes no banco de dados e a busca de clientes existentes utilizando o número de telefone como identificador principal.

```
app.post('/clientes', async (req, res) => {
  const { nome, telefone } = req.body;
  try {
    const result = await pool.query(
      'INSERT INTO clientes (nome, telefone) VALUES ($1, $2) RETURNING id',
      [nome, telefone]
    );
    res.status(201).json({ id: result.rows[0].id });
  } catch (err) {
    console.error(err);
    res.status(500).json({ error: 'Erro ao adicionar cliente' });
  }
});
```

Figura 12 - Criação e Gerenciamento de Clientes

A função `app.post('/clientes')` cria uma rota *POST* para adicionar novos clientes ao banco de dados. Utilizamos SQL parametrizado (`$1`, `$2`) para prevenir injeção de SQL. O `pool.query` é uma função do *PostgreSQL* para executar comandos SQL de forma assíncrona.

4.2 Integração com o WhatsApp

4.2.1 Configuração Inicial e Geração de QR Code

Para integrar o *chatbot* com o WhatsApp, utilizamos a biblioteca *whatsapp-web.js*. A primeira etapa é configurar o cliente e gerar um QR Code para autenticação.

```
const { Client } = require('whatsapp-web.js');
const qrcode = require('qrcode-terminal');
const axios = require('axios');
const client = new Client();

let sessions = {};

client.on('qr', (qr) => {
  qrcode.generate(qr, { small: true });
});

client.on('ready', () => {
  // Cliente está pronto
});
```

Figura 13 - Configuração Inicial e Geração de QR Code

O código acima inicia um novo cliente do WhatsApp. O evento *client.on('qr')* é acionado quando um QR Code é gerado. Usamos *qrcode-terminal* para exibir o QR Code no terminal, que deve ser escaneado pelo aplicativo WhatsApp para iniciar a sessão.

4.2.2 Manipulação de Mensagens e Fluxo de Conversa

O chatbot manipula mensagens recebidas e orienta o usuário por meio de um fluxo de conversa para realizar pedidos ou consultar informações.

```
client.on('message', async (message) => {
  const userId = message.from;
  const text = message.body.trim();
  if (!sessions[userId]) {
    sessions[userId] = { state: 'init', cart: [] };
  }
  const session = sessions[userId];

  try {
    switch (session.state) {
      case 'init':
        await sendMessage(userId, 'Olá! Bem-vindo ao nosso serviço. Qual o seu nome?');
        session.state = 'askName';
        break;
    }
  }
});
```

Figura 14 - Manipulação de Mensagens e Fluxo de Conversa

A função `client.on('message')` escuta mensagens recebidas pelo *chatbot*. Com base no estado atual da sessão (`session.state`), o *chatbot* decide qual mensagem enviar ao usuário. Este controle de fluxo garante uma experiência de usuário fluida e organizada.

4.3 Processamento de Pedidos e Pagamentos

4.3.1 Lógica para Processar Pedidos

A lógica para processar pedidos é uma das partes mais importantes do sistema, pois gerencia a criação de novos pedidos e a interação com o banco de dados.

```
async function finalizeOrder(session, saveOrder) {
  const orderDetails = {
    client_id: session.clientId,
    item: session.selectedItem.name,
    quantity: session.quantity,
    customization: session.customization ? session.customization.description : 'Nenhuma',
    total: calculateTotal(session),
    payment_method: session.paymentMethod,
    address: session.address,
    change: session.change || 0, // Troco, se aplicável
    saveOrder: saveOrder
  };
  try {
    const response = await axios.post('http://localhost:4000/pedidos', orderDetails);
    return response.data;
  } catch (error) {
    console.error('Erro ao finalizar o pedido:', error);
    throw error;
  }
}
```

Figura 15 - Lógica para Processar Pedidos

A função `finalizeOrder` envia uma requisição POST para o servidor para criar um novo pedido no banco de dados. Se o pedido for criado com sucesso, o chatbot confirma o pedido com o usuário; caso contrário, uma mensagem de erro é enviada.

4.4 ChatBot em funcionamento

4.4.1 Usuário Cadastrado Visualizando Pedidos Anteriores

Este teste exemplifica a funcionalidade do *chatbot* ao interagir com um usuário já cadastrado que deseja visualizar seus pedidos anteriores. O fluxo começa com o *chatbot* reconhecendo o número de telefone do usuário, confirmando o cadastro e, em seguida, oferecendo opções para novas ações. Ao escolher a opção de visualizar pedidos anteriores, o *chatbot* busca no banco de dados e retorna uma lista dos pedidos passados, permitindo que o usuário tenha fácil acesso ao histórico de suas compras.

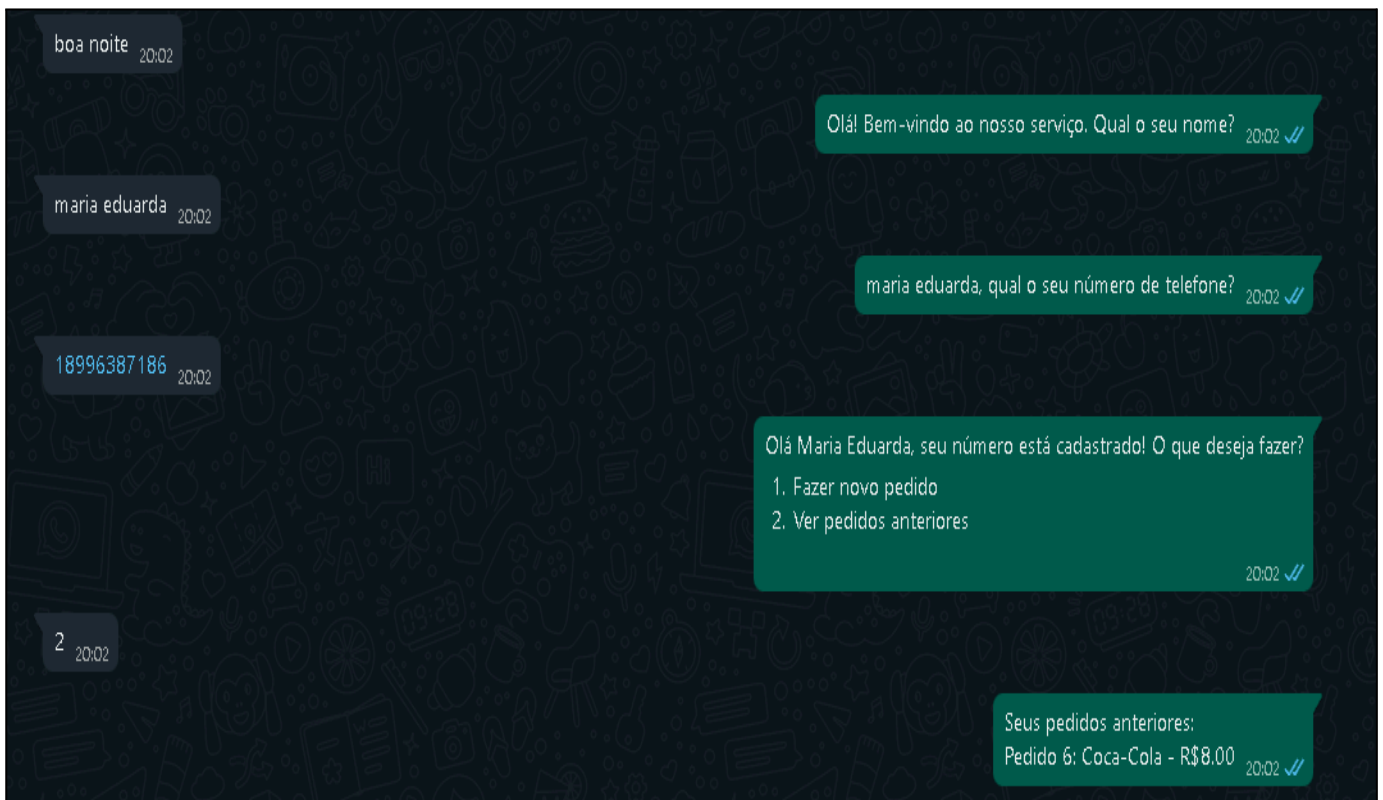


Figura 16 - Teste Usuário Cadastrado Visualizando Pedidos Anteriores

Início da Conversa: O usuário inicia a conversa com uma saudação como "oi" ou "boa noite". O *chatbot* responde com uma mensagem de boas-vindas e solicita o nome do usuário.

Coleta de Informações:

- O *chatbot* pede o nome e o número de telefone do usuário para verificar se o número está cadastrado no sistema.
- Após o usuário fornecer o número de telefone, o *chatbot* faz uma requisição ao banco de dados para verificar se o telefone já está registrado.

Verificação de Cadastro:

- No exemplo da primeira imagem, o número fornecido pelo nome fictício já está cadastrado no sistema. O *chatbot* confirma o cadastro e oferece duas opções: fazer um novo pedido ou visualizar pedidos anteriores.

Exibição de Pedidos Anteriores:

- O usuário seleciona a opção para visualizar pedidos anteriores.
- O *chatbot* realiza uma consulta ao banco de dados utilizando o telefone do cliente para obter o histórico de pedidos associados a esse número.
- Em resposta, o *chatbot* lista os pedidos anteriores, mostrando o ID do pedido e o item adquirido, junto com o preço.

4.4.2 Usuário Fazendo um Pedido e Confirmando o Pagamento

Esse teste demonstra como o *chatbot* gerencia o processo de finalização de um pedido, garantindo que todas as informações essenciais (método de pagamento, endereço de entrega, e detalhes do pedido) sejam corretamente coletadas e confirmadas pelo usuário antes de concluir a transação. Isso assegura uma experiência de compra completa e sem erros para o cliente.

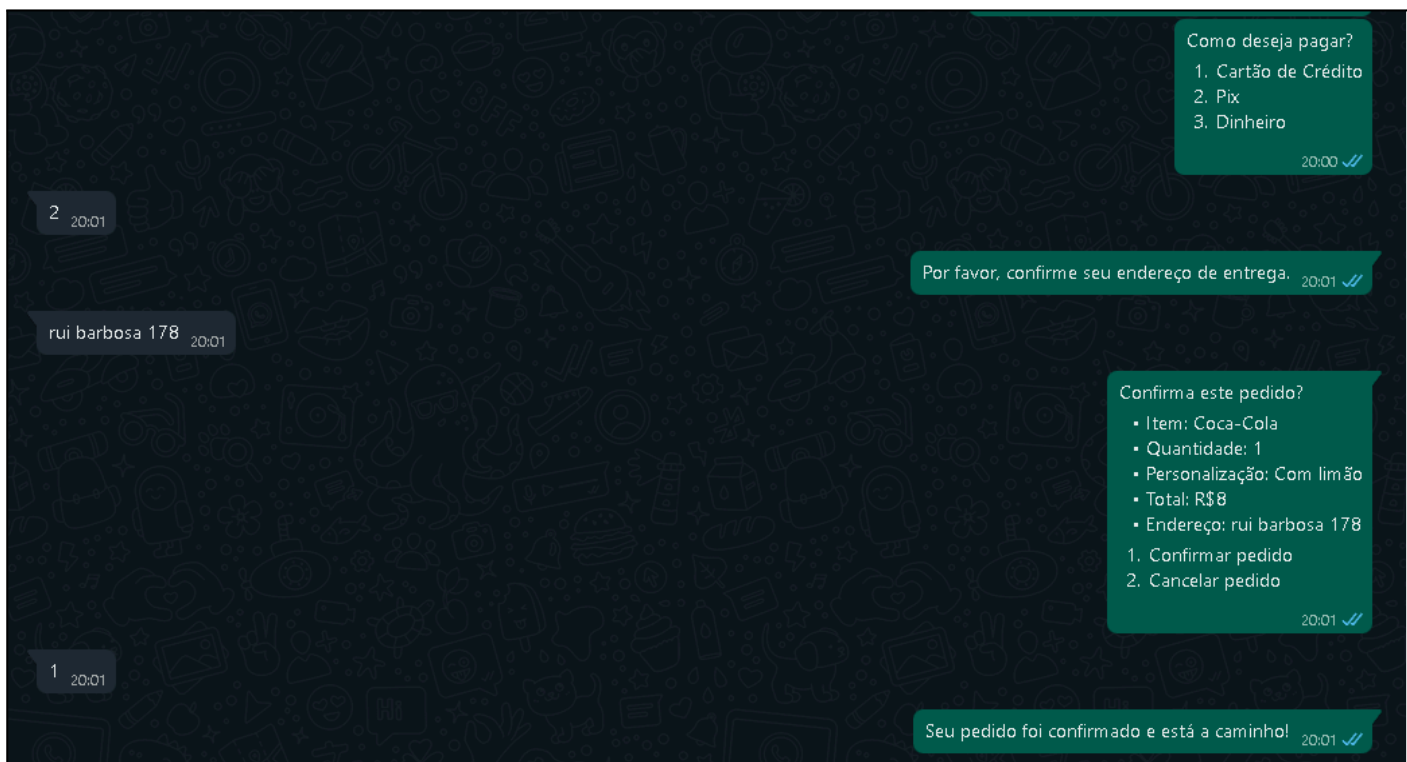


Figura 17 - Teste Usuário Cadastrado Visualizando Pedidos Anteriores

Escolha do Método de Pagamento:

- O *chatbot* solicita que o usuário escolha o método de pagamento após selecionar o item e a quantidade desejada. As opções são "Cartão de Crédito", "Pix", e "Dinheiro".
- No exemplo, o usuário escolhe a opção "2" para pagamento via Pix.

Confirmação do Endereço:

- Após a escolha do método de pagamento, o *chatbot* pede para o usuário confirmar o endereço de entrega. Este passo é crucial para garantir que o pedido seja enviado para o local correto.
- O usuário fornece o endereço "rui barbosa 178".

Resumo e Confirmação do Pedido:

- O *chatbot* então exibe um resumo detalhado do pedido, incluindo o item selecionado (Coca-Cola), a quantidade (1), a personalização (Com limão), o total do pedido (R\$8), e o endereço de entrega informado pelo usuário.
- O *chatbot* pede ao usuário para confirmar o pedido ou cancelá-lo, oferecendo as opções "1. Confirmar pedido" e "2. Cancelar pedido".

Confirmação Final:

- O usuário escolhe a opção "1" para confirmar o pedido.
- O *chatbot* responde confirmando que o pedido foi processado e está a caminho.

5. CONCLUSÕES

O desenvolvimento do *chatbot* para o atendimento ao cliente no setor de restaurantes destaca-se como uma aplicação eficiente da tecnologia em um ambiente cada vez mais voltado para o digital. A solução proposta facilita o processo de pedidos, gerenciamento de histórico de compras e interação personalizada com os clientes, utilizando tecnologias como JavaScript, Node.js, PostgreSQL e a API do WhatsApp.

Por meio deste sistema, o restaurante consegue não apenas otimizar seu atendimento, mas também oferecer uma experiência mais conveniente e direta ao cliente, aumentando a satisfação e a eficiência operacional. Este trabalho demonstra que a adoção de chatbots inteligentes é uma tendência cada vez mais relevante para negócios que buscam melhorar a interação com seus clientes e se adaptar às demandas de um mercado em constante evolução.

5.1 TRABALHOS FUTUROS

O sistema desenvolvido, apesar de já melhorar o atendimento ao cliente, tem um grande potencial para futuras expansões e aprimoramentos. Para os próximos passos, pode-se considerar a implementação de algoritmos de machine learning que permitam a personalização automática dos serviços, prevendo preferências dos clientes a partir do histórico de interações e comportamentos de compra. Além disso, a expansão do suporte para múltiplas plataformas de comunicação, incluindo redes sociais e assistentes virtuais, permitiria um atendimento mais abrangente e acessível.

Outras melhorias potenciais incluem o desenvolvimento de um painel administrativo mais intuitivo para facilitar o gerenciamento de cardápios, promoções e inventários em tempo real, bem como a integração com sistemas de pagamento diversificados para aumentar a conveniência do cliente. Adicionalmente, a implementação de análises automatizadas de feedbacks dos clientes pode fornecer insights estratégicos para ajustes contínuos nos serviços oferecidos, aprimorando ainda mais a experiência do usuário e a eficiência operacional. Essas evoluções promoveriam um uso mais sofisticado de tecnologias no setor de hospitalidade, elevando o padrão de atendimento e gestão.

REFERÊNCIAS

ALURA. Node.js: o que é, como funciona e por que utilizá-lo? ALURA, 2023. Disponível em: <https://www.alura.com.br/artigos/node-js?srsIid=AfmBOooUNOAz4tDwNfkegy-agF8mlbeqXuu6r5nlOLSLr9mtbylBIY44>. Acesso em: 4 ago. 2024.

AQUINO, Victor; ADANIYA , Mario. **DESENVOLVIMENTO E APLICAÇÕES DE CHATBOT.** CHATBOT DEVELOPMENT AND APPLICATIONS, [s. l.], 31 jul. 2018. Acesso em: 16 nov. 2023.

BARROS, Daniela; GUERREIRO, Aníbal. **Novos desafios da educação a distância: programação e uso de Chatbots.** [s. l.], 12 dez. 2018. Disponível em: <https://repositorioaberto.uab.pt/bitstream/10400.2/8410/1/8743-Texto%20do%20artigo-15295536-1-10-20190510.pdf>. Acesso em: 16 nov. 2023.

BOOCH, Grady; JACOBSON, Ivar; RUMBAUCH, James. **UML – Guia do Usuário.** 2ª Edição. Tradução Fábio Freitas da Silva e Cristiana de Amorim Machado. Rio de Janeiro: Elsevier, 2005. Acesso em: 20 nov. 2023.

FOWLER, Martin. UML Essencial. Grupo A, 2011. 9788560031382. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788560031382/>. Acesso em: 8 mar. 2024.

GHIDINI, ITAMAR; MATTOS, WINICIUS. **DESENVOLVIMENTO E APLICAÇÃO DE UM CHATBOT PARA AUXILIAR O ATENDIMENTO AO CLIENTE.** 2018. UNIVERSIDADE DO SUL DE SANTA CATARINA, [S. l.], 2018.

JÚNIOR, Ciro; CARVALHO, Kely. **VISÃO GERAL SOBRE APLICAÇÕES INTELIGENTES.** 2018. Instituto Federal do Tocantins, [S. l.], 2018. Acesso em: 15 nov. 2023.

LUGLI, Erônica Adelaide; FILHO , João de Lucca. **O USO DO CHATBOT PARA A EXCELÊNCIA EM ATENDIMENTO.** 2020. 14 p. Faculdade de Tecnologia de Taquaritinga, Taquaritinga, 2020.

SOUZA, Ivan. **PostgreSQL: saiba o que é, para que serve e como instalar.** Rock Content. Disponível em: <<https://rockcontent.com/br/blog/postgresql/>>. Acesso em: 23 nov. 2023.

XP EDUCAÇÃO. O que é Express JS e qual sua função? XP EDUCAÇÃO, 2023. Disponível em: <https://blog.xpeducacao.com.br/express-javascript/>. Acesso em: 29 ago. 2024.