Desenvolvimento de um Chatbot Conversacional para Seleção de Mármores e Granitos

Raul Camargo Floriano Ribeiro
CPCX–UFMS
Coxim, MS, Brasil
raul.camargo@ufms.br

Ekler Paulino de Mattos
CPCX–UFMS
Coxim, MS, Brasil
ekler.mattos@ufms.br

RESUMO

Este trabalho apresenta o MarmoAI, um chatbot conversacional desenvolvido para ajudar na recomendação e escolha de mármores e granitos. Basicamente, o MarmoAI é baseado em Processamento de Linguagem Natural (PLN) com uma base de conhecimento estruturada em PDF, permitindo oferecer respostas tanto precisas quanto personalizadas para cada usuário. O projeto foi estruturado com Python e Flask como base do sistema, e utilizado o Twilio para fazer a integração com WhatsApp – essa escolha se mostrou fundamental para garantir que a interação fosse realmente fluida e de fácil acesso. Para mensurar os resultados foi utilizada a métrica perplexity, amplamente utilizada na literatura em modelos de linguagem [4], onde foi obtido uma coerência textual de 94,08. Foi aplicado também uma avaliação de satisfação do usuário com Chatbot. Os resultados obtidos comprovam não apenas a viabilidade técnica da proposta, mas também evidenciam o potencial impacto que sistemas conversacionais como este podem ter no setor de rochas ornamentais, um segmento que tradicionalmente ainda depende muito de consultas presenciais e catálogos físicos.

KEYWORDS

Chatbot, Mármores, Granitos, Processamento de Linguagem Natural, Inteligência Artificial

1 INTRODUÇÃO

As marmorarias representam um elo fundamental entre a extração de pedras naturais e sua aplicação final em obras de construção civil e projetos de design de interiores. O mercado de mármores e granitos movimenta valores expressivos no Brasil, com aplicações que vão desde pequenas reformas em residências até grandes projetos comerciais. Quando um cliente escolhe trabalhar com essas pedras naturais, os motivos vão além da aparência: há também a questão da resistência do material e do quanto ele pode valorizar o imóvel a longo prazo. Acontece que, mesmo com toda essa demanda, o atendimento nesse setor ainda apresentam falhas. Comumente consumidores cheguam às lojas sem saber exatamente qual opção de pedra melhor atende as suas necessidades — alguns não têm conhecimento técnico suficiente sobre as diferenças entre os materiais. Por outro lado, outros encontram dificuldade em conseguir orientação adequada dos vendedores, especialmente em horários de maior movimento. Foi pensando nesse problema que surgiram os chatbots baseados em inteligência artificial como uma solução prática e acessível.

Este trabalho propõe o desenvolvimento do MarmoAI, um chatbot projetado especificamente para atuar como consultor virtual na escolha de mármores e granitos. O sistema foi concebido com dois objetivos principais: tornar o processo de seleção mais acessível para o público leigo e, simultaneamente, reduzir a sobrecarga sobre as equipes de atendimento das marmorarias. Do ponto de vista metodológico, optou-se por uma arquitetura modular que combina recursos de Processamento de Linguagem Natural com uma base documental em PDF contendo especificações técnicas dos materiais. Essa escolha permitiu que o chatbot processasse consultas em linguagem natural e retornasse recomendações fundamentadas em informações confiáveis. A validação envolveu dois eixos principais: testes práticos de usabilidade com usuários reais e análises quantitativas sobre a precisão das respostas geradas. Foi conduzido avaliações quantitativas por meio de um questionário de satisfação e de métricas automáticas, como perplexity. O questionário indicou boa aceitação do sistema, tanto em clareza das respostas quanto em utilidade percebida. Já os testes de desempenho mostraram que o MarmoAI respondeu às interações avaliadas em um tempo total de aproximadamente 25,8 segundos, com média de 3,23 segundos por resposta. As mensagens geradas apresentaram em média 289 tokens e 490 caracteres, mantendo um nível de detalhamento adequado ao usuário final. Quanto à métrica de perplexity, obteve-se média de 160,86, variando entre 94,08 (maior coerência) e 204,77 (menor coerência), o que demonstra um nível de consistência textual considerado satisfatório, especialmente em respostas que exigem explicações técnicas mais extensas.

2 REVISÃO DA LITERATURA

Diversos estudos têm explorado o uso de Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN) em sistemas de recomendação e atendimento ao cliente. A seguir, detalhamos algumas das contribuições relevantes.

2.1 Chatbots para Recomendação e Atendimento ao Cliente

Jain and Soni [3] exemplificam a eficácia de um chatbot na recomendação de materiais e para isso eles desenvolveram uma solução chamada MarbleBot. O MarbleBot foi desenvolvido para auxiliar clientes na seleção de mármores e granitos, integrando filtragem baseada em conteúdo e técnicas de PLN. A principal tecnologia utilizada foi o Dialogflow usada para a construção da interface conversacional, complementada por uma API Flask e por fim um banco de dados MySQL para gerenciar as informações dos produtos e interações. Os resultados coletados demonstraram uma taxa de precisão de até 85% nas recomendações e uma alta satisfação por parte do usuário, assim, provando ser uma ferramenta valiosa para personalizar a experiência do cliente e otimizar o processo de seleção.

Clement [2], explora as diversas contribuições dos chatbots de inteligência artificial para o aprimoramento da experiência do cliente. O autor ressalta, em particular, como essas ferramentas se mostram eficazes na agilização da resolução de consultas, na oferta de um **suporte ininterrupto (24/7)**, na redução dos custos operacionais e no incremento do engajamento do cliente. Por meio de uma análise aprofundada da literatura e de diversos estudos de caso, Clement conclui que a incorporação de chatbots de IA não apenas eleva a qualidade do serviço prestado, mas também confere uma vantagem competitiva crucial para a fidelização e retenção de clientes. Para que essa integração seja bem-sucedida, fatores como responsividade, precisão, uma certa dose de inteligência emocional e a perfeita harmonização com os processos existentes de atendimento ao cliente são considerados essenciais.

Ok [5] investiga como os chatbots de IA impactam a lealdade do cliente. O autor explora como esses chatbots ajudam e aprimoram a experiência do usuário, melhoram a eficiência das respostas e personalizam o engajamento do cliente, resultando em uma maior satisfação e retenção. Este estudo analisa fatores como a capacidade de resposta do chatbot, precisão, inteligência emocional e integração com as operações de atendimento. O autor conclui que os chatbots de IA são ferramentas estratégicas para construir confiança, otimizar operações e fortalecer a lealdade à marca em mercados digitais competitivos.

Rahulil et al. [6] focaram no desenvolvimento e a aplicação de chatbots no ambiente do WhatsApp. Valendo-se da metodologia PRISMA, a pesquisa revelou que as abordagens baseadas em PLN (Processamento de Linguagem Natural) são as mais frequentemente empregadas na criação desses assistentes virtuais. Além disso, O autor mencionou que a linguagem Python surge como uma linguagem de programação predominante, um reflexo de sua notável flexibilidade e do grande ecossistema de bibliotecas robustas que essa linguagem oferece (a exemplo de NLTK, spaCy e TensorFlow). O trabalho também aponta a ampla aplicabilidade desses chatbots em diversos setores, tais como saúde, negócios e educação. O estudo termina ressaltando desafios e promissoras oportunidades futuras, incluindo a integração de funcionalidades adicionais e o aprimoramento da capacidade de compreensão do contexto conversacional.

No contexto de IA na Indústria de Rochas Ornamentais, a revisão sistemática da literatura proposta por Silva et al. [7], enfatiza o potencial da IA na indústria de rochas ornamentais. O estudo revisa as tendências no uso de técnicas de IA e Machine Learning (ML) na indústria entre 2017 e 2024. A metodologia de Revisão Sistemática da Literatura em ciência da computação revelou que uma vasta pesquisa tem sido conduzida na classificação de ladrilhos, com soluções robustas para aplicação industrial. Outros tópicos abordados incluem corte de pedra, detecção de defeitos, previsão de variáveis e monitoramento de atividades de pedreiras. Os autores sugerem que, embora haja necessidade de mais pesquisas em áreas específicas, o trabalho oferece um ponto de partida sólido para futuras investigações.

2.2 Contribuição do Presente Trabalho

Em contraste com os estudos existentes, que abordam a aplicação de IA e PLN em contextos gerais de recomendação e atendimento ou em aspectos específicos da indústria de rochas ornamentais (como classificação e detecção de defeitos), o presente trabalho foca no desenvolvimento de um sistema de recomendação conversacional adaptado para o setor de rochas ornamentais, com um diferencial na sua validação. Enquanto trabalhos como o proposto por Jain and Soni [3] demonstram a viabilidade de chatbots para recomendação de materiais.

Clement [2] e Ok [5] exploram o impacto dos chatbots na satisfação e lealdade do cliente através de questionários sobre a aplicação. Em contrapartida, a nossa proposta visa não apenas aprimorar a experiência do usuário, mas também medir a precisão da resposta obtida pelo chatbot. Além do uso de questionários tradicionais para avaliar a satisfação, a solução proposta incorpora métricas específicas para quantificar a qualidade da resposta, diferenciando-se assim pela profundidade da avaliação da eficácia educacional do sistema. Isso permitirá uma compreensão mais completa do valor agregado do chatbot, indo além da simples recomendação.

3 METODOLOGIA

O desenvolvimento do chatbot seguiu uma abordagem modular, focando na integração de componentes para processamento de linguagem natural, gerenciamento de base de conhecimento e comunicação com o usuário via WhatsApp. A arquitetura do sistema é composta por:

► **Base de Conhecimento**

A base de conhecimento do chatbot é um arquivo PDF chamado **manual_marmoraria.pdf** que contêm informações detalhadas sobre mármores, granitos, suas características, aplicações e cuidados. Este documento foi lido e seu conteúdo extraído programaticamente para ser utilizado como contexto nas interações com a IA. A escolha de um PDF como base de conhecimento permite fácil atualização e manutenção das informações.

► **Processamento de Linguagem Natural (PLN)**

Para o PLN, foi utilizada uma API de modelo de linguagem (DeepSeek via OpenRouter[1]), que permite ao chatbot compreender as perguntas dos usuários e gerar respostas coerentes. A escolha do DeepSeek se deu por seu destaque em testes, demonstrando ser uma opção gratuita, rápida e consistente nas respostas, superando outras alternativas avaliadas. A API é configurada para receber o histórico da conversa e o conteúdo relevante do PDF, garantindo que as respostas sejam contextuais e baseadas na base de conhecimento fornecida. A integração com a API é feita através de requisições HTTP POST.

► **Integração com WhatsApp**

A comunicação com o WhatsApp é realizada através da plataforma Twilio[2], que atua como uma ponte entre o chatbot e os usuários. Um webhook — um mecanismo que permite que o Twilio notifique automaticamente a aplicação quando uma nova mensagem chega — é configurado para direcionar as mensagens recebidas para uma aplicação Flask[3]. Esta aplicação, construída com o microframework web Flask para Python, é responsável por receber a notificação, processar a mensagem, interagir com o modelo de linguagem e enviar a resposta de volta ao Twilio no formato TwiML

[1] https://openrouter.ai/deepseek
[2] https://www.twilio.com/whatsapp
[3] https://flask.palletsprojects.com/

(Twilio Markup Language), que então a encaminha para o WhatsApp do usuário.

Para melhor entendimento, segue a representação visual desta arquitetura, apresentada na figura 1. De forma simplificada, o fluxo representado na figura inicia quando o usuário envia uma mensagem pelo WhatsApp. Essa mensagem é encaminhada ao Twilio, que, por meio do webhook configurado, realiza uma requisição HTTP para a aplicação desenvolvida em Flask. A aplicação Flask, ao receber essa requisição, processa o conteúdo da mensagem, consulta o modelo de linguagem juntamente com a base de conhecimento em PDF e gera uma resposta adequada. Em seguida, essa resposta é enviada de volta ao Twilio no formato TwiML, que por sua vez a encaminha novamente ao WhatsApp, retornando-a ao usuário de forma transparente.

3.1 Métricas para a Validação da Proposta

Para validar a proposta, realizaram-se análises quantitativas, baseadas em métricas de desempenho, e qualitativas, por meio de questionários de satisfação aplicados aos usuários. Os procedimentos de avaliação são apresentados a seguir.

► **A métrica Perplexity**

Neste trabalho utilizamos a métrica *perplexity* para avaliar o desempenho do modelo. Particularmente, o *perplexity* é uma métrica utilizada na avaliação de modelos de linguagem, especialmente em modelos de aprendizado profundo (LLMs), para medir a eficácia do modelo em prever uma sequência de palavras [1].

A métrica *perplexity* usada neste trabalho segue a definição tradicional apresentada por Jurafsky e Martin [4]. Nesse método, a perplexidade é calculada a partir da perda (*loss*) do modelo de linguagem. Para realizar esse cálculo, foi utilizada a biblioteca HuggingFace Transformers com o modelo GPT-2, que permite medir o nível de coerência e fluência do texto de forma automática.

► **Tempo de Resposta vs. Tamanho da Resposta**

Outro aspecto analisado neste trabalho foi a relação entre o tempo de resposta do sistema e o tamanho das respostas geradas, medido em número de caracteres. Essa métrica tem por objetivo avaliar o impacto do volume de conteúdo na latência de geração da resposta, um fator importante para a experiência do usuário, especialmente em sistemas interativos.

► **Avaliação de Satisfação do Usuário**

Além das análises quantitativas com as as métricas de desempenho, também realizamos uma análise qualitativa a partir de um questionário de satisfação a fim de avaliar a experiência do usuário na interação com o chatbot. Para isso, elaboramos um questionário composto por um conjunto de perguntas destinadas a medir a clareza, a relevância e a utilidade das respostas fornecidas pelo sistema, bem como o nível geral de satisfação do usuário.

O questionário foi dividido em blocos correspondentes a cada uma das sete perguntas, apresentadas na Tabela 1, previamente definidas para avaliação do chatbot. Para cada interação, o participante foi convidado a informar:

- Se a resposta gerada pelo chatbot foi relevante para a pergunta apresentada;
- Se a resposta fez sentido e foi fácil de entender;
- Se o nível de satisfação geral com aquela resposta, indicado por uma escala de 1 a 5, onde 1 representa baixa satisfação e 5 representa satisfação máxima.

Ao final do formulário, foi disponibilizado um campo opcional para comentários adicionais, permitindo que os usuários oferecessem sugestões ou observações livres sobre o funcionamento do sistema.

4 MARMOAI

O MarmoAI é um sistema conversacional composto por dois módulos principais que trabalham de forma integrada para fornecer assistência especializada na seleção de mármores e granitos. A arquitetura do sistema é definida pelos Algoritmos 1 e 2, que detalham, respectivamente, o processamento conversacional e a integração com WhatsApp.

Algorithm 1: Algoritmo de processamento conversacional do MarmoAI.

Data: msg_{user}; PDF_{base}; $history_{user}$
Result: $response_{AI}$
// Inicialização do sistema
if $history_{user} = \emptyset$ **then**
  $content_{PDF} \leftarrow$ extractPDF(PDF_{base});
  $prompt_{system} \leftarrow$ createSystemPrompt($content_{PDF}$);
  $history_{user} \leftarrow [prompt_{system}]$;
end
// Processamento da mensagem do usuário
$history_{user}$.append(msg_{user});
// Comunicação com modelo de linguagem
$payload \leftarrow$ { model: "deepseek", messages: $history_{user}$, temperature: 0.5 };
$response_{API} \leftarrow$ POST(OpenRouter_API, $payload$);
if $response_{API}.status = 200$ **then**
  $response_{AI} \leftarrow response_{API}$.content;
  $history_{user}$.append($response_{AI}$);
  // Cálculo de métricas de qualidade
  $perplexity \leftarrow$ calculatePerplexity($response_{AI}$);
  $metrics \leftarrow$ { perplexity: $perplexity$, tokens: count($response_{AI}$), time: $t_{response}$ };
  saveMetrics($metrics$);
end
else
  $response_{AI} \leftarrow$ "Erro: tente novamente";
end
return $response_{AI}$;

4.1 Algoritmo de Processamento Conversacional

O Algoritmo 1 descreve o fluxo principal de processamento do MarmoAI. O algoritmo recebe como entrada a mensagem do usuário (msg_{user}), a base de conhecimento em PDF (PDF_{base}) e o histórico de conversas do usuário ($history_{user}$), produzindo como saída a resposta gerada pela IA ($response_{AI}$).

O processamento inicia com a verificação do histórico conversacional. Caso seja a primeira interação do usuário ($history_{user} = \emptyset$),

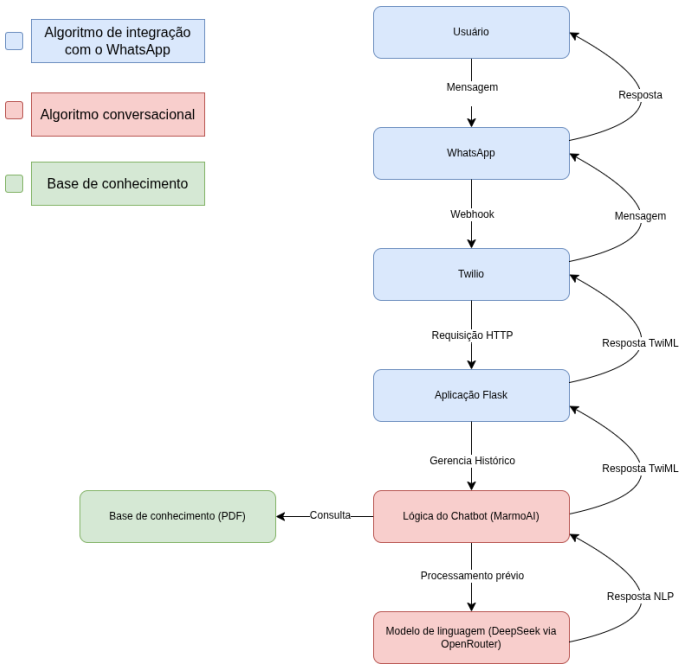

Figura 1: Arquitetura do sistema do chatbot, detalhando o fluxo de comunicação e processamento.

o sistema realiza três operações fundamentais: extrai o conteúdo textual do PDF utilizando a função `extractPDF`, constrói um *prompt* de sistema que contextualiza o papel do assistente e incorpora o conhecimento extraído através da função `createSystemPrompt`, e inicializa o histórico com este *prompt* de sistema (linhas 1-5 do Algoritmo 1).

Após a inicialização, a mensagem do usuário é adicionada ao histórico conversacional. O sistema então prepara uma estrutura de dados (*payload*) contendo o modelo de linguagem a ser utilizado (DeepSeek), o histórico completo de mensagens e parâmetros de geração (temperatura de 0.5 para respostas mais determinísticas). Esta estrutura é enviada via requisição HTTP POST para a API do OpenRouter (linhas 6-8).

Ao receber a resposta da API, o algoritmo verifica o status da requisição. Em caso de sucesso (status 200), a resposta da IA é extraída e adicionada ao histórico conversacional. Subsequentemente, o sistema calcula métricas de qualidade através da função `calculatePerplexity`, que avalia a coerência e naturalidade da resposta gerada. Essas métricas, incluindo perplexidade, número de tokens e tempo de resposta, são armazenadas para análise posterior (linhas 13 e 14). Caso ocorra falha na comunicação com a API, uma mensagem de erro padronizada é retornada ao usuário.

4.2 Algoritmo de Integração com WhatsApp

O Algoritmo 2 detalha a integração do MarmoAI com a plataforma WhatsApp através do serviço Twilio. O algoritmo recebe como entrada os dados do webhook Twilio ($webhook_{twilio}$) e a instância do chatbot ($chatbot_{instance}$), produzindo como saída a resposta formatada em TwiML ($response_{TwiML}$).

O processamento inicia com a extração de dois elementos essenciais do webhook: o número de telefone do remetente ($phone_{number}$), que serve como identificador único do usuário, e o corpo da mensagem ($msg_{incoming}$). O sistema mantém um dicionário de históricos (*histories*) que associa cada número de telefone ao seu respectivo histórico conversacional (linhas 1 e 2 do Algoritmo 2).

Caso seja a primeira interação de um usuário específico (i.e., $phone_{number} \notin histories$), um novo histórico é criado e inicializado com o *prompt* de sistema (linhas 3-5). Isso garante que cada usuário tenha uma sessão conversacional independente e contextualizada. O histórico correspondente ao usuário atual é então recuperado e atribuído à instância do chatbot, estabelecendo o contexto da conversa (linhas 6 e 7).

A mensagem recebida é processada através do método `sendMessage` do chatbot, que invoca o Algoritmo 1 internamente (linha 8). Após o processamento, o histórico atualizado (incluindo a nova mensagem do usuário e a resposta da IA) é armazenado de volta no dicionário de históricos, preservando o contexto para interações futuras (linha 9).

Por fim, a resposta da IA é formatada no padrão TwiML (*Twilio Markup Language*) através da função `formatTwiML`, que estrutura a mensagem no formato XML esperado pelo Twilio para envio via WhatsApp (linha 10). Esta resposta formatada é então retornada ao webhook, completando o ciclo de comunicação (linha 11).

4.3 Implementação dos Módulos

4.3.1 `marmoAIPDF.py`. Este arquivo implementa o Algoritmo 1, contendo a lógica central do chatbot através da classe `MarmoraiaChatbot`. Suas principais funcionalidades incluem:

- **Carregamento da base de conhecimento**: Utiliza a biblioteca PyPDF2 para extrair o conteúdo textual do PDF e armazená-lo em memória.
- **Inicialização conversacional**: Cria um prompt de sistema que define o papel do assistente como especialista em marmoraria e incorpora as primeiras 8000 caracteres da base de conhecimento.
- **Comunicação com API**: Gerencia as requisições HTTP para a API OpenRouter, incluindo tratamento de erros e timeout.
- **Cálculo de métricas**: Implementa a classe `PerplexityCalculator` utilizado para calcular a perplexidade das respostas, uma métrica que quantifica a naturalidade e coerência do texto gerado.
- **Geração de relatórios**: Coleta e armazena métricas de todas as interações, gerando relatórios detalhados em formato JSON para análise de desempenho.

4.3.2 `twilio_flask_adapter.py`. Este arquivo implementa o Algoritmo 2, atuando como adaptador entre o Twilio e o chatbot. Utiliza o framework Flask para criar um servidor web com as seguintes responsabilidades:

- **Endpoint webhook**: Define a rota `/twilio_webhook` que recebe requisições POST do Twilio contendo as mensagens dos usuários.
- **Gerenciamento de sessões**: Mantém um dicionário em memória (*histories*) que armazena o histórico conversacional de cada usuário, identificado pelo número de telefone.
- **Integração com chatbot**: Configura a instância do `Marmoraria Chatbot` com o histórico apropriado antes de processar cada mensagem.
- **Formatação TwiML**: Utiliza a biblioteca `twilio.twiml.messaging_response` para construir respostas XML válidas que o Twilio pode interpretar e encaminhar ao WhatsApp.

Algorithm 2: Algoritmo de integração WhatsApp via Twilio.

Data: $webhook_{twilio}$; $chatbot_{instance}$
Result: $response_{TwiML}$
// Recepção da mensagem via WhatsApp
$phone_{number} \leftarrow webhook_{twilio}$.From;
$msg_{incoming} \leftarrow webhook_{twilio}$.Body;
// Recuperação ou criação do histórico
if $phone_{number} \notin histories$ **then**
  $histories[phone_{number}] \leftarrow$ [systemPrompt];
end
$history_{current} \leftarrow histories[phone_{number}]$;
// Configuração do contexto do chatbot
$chatbot_{instance}$.conversation_history $\leftarrow history_{current}$;
// Processamento da mensagem
$response_{AI} \leftarrow chatbot_{instance}$.sendMessage($msg_{incoming}$);
// Atualização do histórico
$histories[phone_{number}] \leftarrow$ $chatbot_{instance}$.conversation_history;
// Formatação da resposta para WhatsApp
$response_{TwiML} \leftarrow$ formatTwiML($response_{AI}$);
return $response_{TwiML}$;

5 RESULTADOS E DISCUSSÃO

Para avaliar o desempenho do chatbot desenvolvido, foram realizadas sete perguntas comuns sobre o uso de mármores e granitos, de forma a simular dúvidas reais de clientes, conforme apresentado na Tabela 1. As respostas foram processadas pelo modelo linguístico integrado ao sistema e analisadas com base em métricas quantitativas de desempenho, conforme apresentadas na Tabela 2.

Tabela 1: Perguntas utilizadas para avaliação do chatbot

#	Pergunta
1	O mármore pode ser usado em cozinhas?
2	O granito é mais resistente do que o mármore?
3	Como fazer a manutenção do mármore no dia a dia?
4	O que mancha mais facilmente, mármore ou granito?
5	Quais cuidados devo ter logo após a instalação do mármore?
6	Qual a diferença entre acabamento polido, levigado e escovado?
7	Posso usar granito em áreas externas?

5.1 Desempenho geral do modelo

O tempo total de resposta para as sete interações realizadas foi de aproximadamente 25,8 segundos, com uma média de 3,23 segundos por resposta (vide Tabela 2). As respostas tiveram em média 289 tokens e 490 caracteres, indicando uma extensão textual adequada à comunicação com o usuário final.

A métrica de perplexity apresentou uma média de 160,86, com valores variando entre 94,08 (melhor coerência textual) e 204,77 (menor coerência). Esses resultados demonstram que o sistema apresenta um nível de consistência textual considerado satisfatório, especialmente em perguntas que envolvem explicações detalhadas ou procedimentos técnicos.

5.2 Análise individual das respostas com Perplexity

A Figura 2 mostra a distribuição da *perplexity* para cada uma das perguntas testadas. Nota-se que perguntas mais descritivas e com maior volume de texto, como "Como fazer a manutenção do mármore no dia a dia?", apresentaram valores de *perplexity* mais baixos (94,08), refletindo maior coerência linguística e estabilidade do modelo. Já respostas curtas e diretas, como "O que mancha mais facilmente, mármore ou granito?", tiveram *perplexity* mais alta (204,77).

De modo geral, os resultados indicam que o MarmoAI apresenta qualidade textual **boa à excelente**, conforme a interpretação das faixas de *perplexity*. O valor médio de 160,86 situa o desempenho do sistema na faixa *razoável*, próximo à fronteira de qualidade boa, evidenciando coerência semântica e estabilidade na geração das respostas. Esse comportamento é esperado, uma vez que valores mais altos de perplexity indicam maior dificuldade do modelo em prever a sequência textual, o que reflete menor coerência e maior incerteza no processo de geração.

Tabela 2: Resumo das métricas gerais do chatbot

Métrica	Valor Total	Médio	Mínimo	Máximo
Tempo de resposta (s)	25.83	3.23	1.98	5.45
Perplexity	—	160.86	94.08	204.78
Número de tokens	—	192	75	330
Tamanho da resposta (caracteres)	—	490	191	814

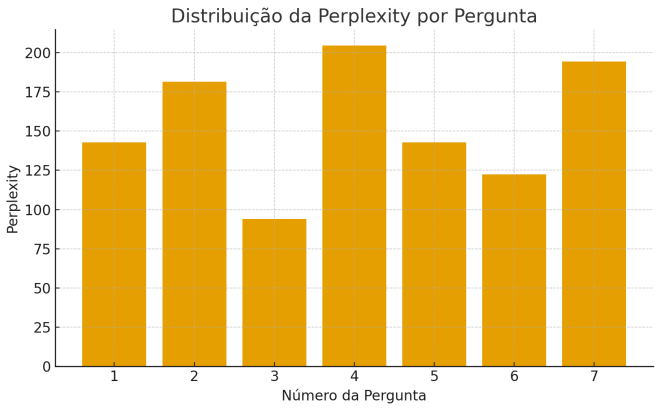

Figura 2: Distribuição da *perplexity* por pergunta avaliada

5.3 Tempo de Resposta e Tamanho da Resposta

Outro aspecto observado foi a relação entre o tempo de resposta e o tamanho da resposta, como ilustrado na Figura 3. Houve uma correlação positiva: respostas mais longas demandaram maior tempo de processamento, embora o sistema tenha se mantido abaixo de 6 segundos em todos os casos, o que garante fluidez na interação.

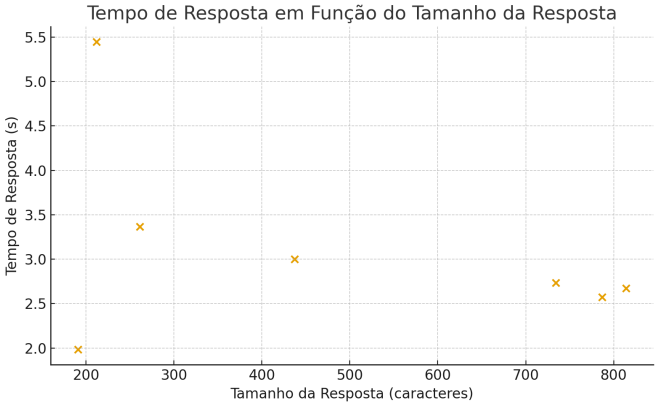

Figura 3: Tempo de resposta em função do tamanho das respostas

5.4 Avaliação de Satisfação do Usuário

Na análise qualitativa, a solução proposta foi apresentada a três empresas do setor de comércio de mármore. Em seguida, aplicou-se o questionário descrito na Seção 3.1. A Figura 4 apresenta os níveis de satisfação atribuídos pelos participantes para cada pergunta respondida pelo chatbot. Nota-se que as avaliações se concentram majoritariamente nos valores mais altos da escala, indicando uma percepção globalmente positiva sobre a qualidade das respostas fornecidas. A consistência entre as diferentes perguntas evidencia que o chatbot manteve um bom desempenho independentemente do tema abordado.

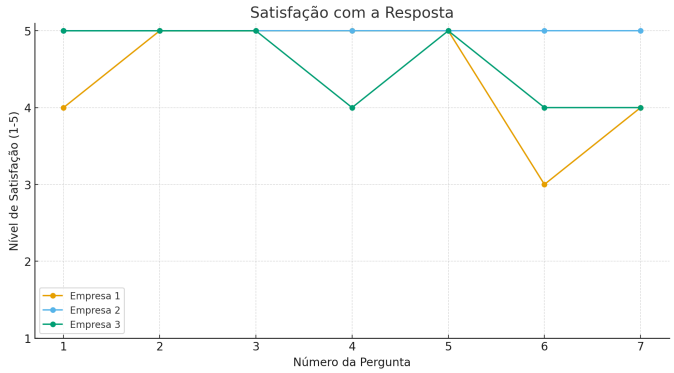

Figura 4: Níveis de satisfação por pergunta e por participante.

Figura 5: Se a resposta gerada pelo chatbot fez sentido e foi fácil de entender

A Figura 5 mostra a percepção dos usuários sobre a clareza das respostas fornecidas pelo chatbot. Todos os participantes indicaram que as respostas fizeram sentido e foram fáceis de entender, o que demonstra que o sistema apresentou comunicação clara, objetiva e acessível.

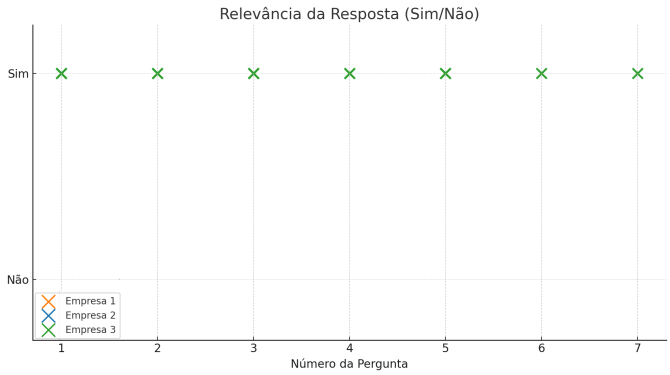

Figura 6: Se a resposta gerada pelo chatbot foi relevante para a pergunta.

A Figura 6 ilustra o grau de relevância atribuído às respostas do chatbot em relação às perguntas realizadas. Os resultados revelam unanimidade na avaliação positiva, confirmando que o chatbot interpretou corretamente as demandas e forneceu informações pertinentes.

5.5 Demonstração de Funcionamento

Para ilustrar a aplicação prática do chatbot em um cenário de uso real, na Figura 7 apresenta uma captura de tela de uma conversa no WhatsApp. Nesta interação, um usuário pergunta simultaneamente qual o melhor material para uma bancada de cozinha e qual o número de contato da marmoraria, e a resposta cita o número de telefone (67) 9999-9999, que é o número de telefone fornecido na base de conhecimento PDF.

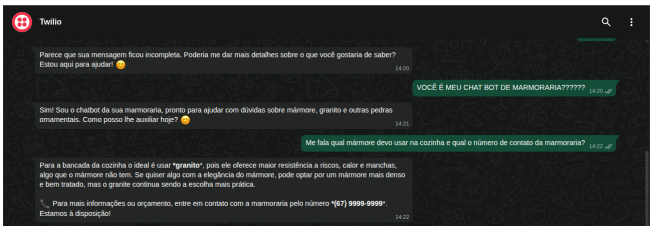

Figura 7: Exemplo de interação com o chatbot no WhatsApp.

6 CONCLUSÃO

Este trabalho apresentou o MarmoAI: um chatbot conversacional eficaz para o setor de mármores e granitos, utilizando uma abordagem prática e acessível. A integração de uma base de conhecimento em PDF com um modelo de linguagem via API e a comunicação com o WhatsApp através de Flask e Twilio resultaram em uma ferramenta robusta e funcional. O projeto superou desafios técnicos e validou a capacidade da IA em fornecer informações precisas e personalizadas, reforçada pelo melhor valor de *perplexity* obtido (94,08).

Como trabalhos futuros, sugere-se aprimorar a capacidade da IA para nuances e gírias, explorar a integração com outros canais de comunicação e desenvolver uma interface de gerenciamento para a base de conhecimento, facilitando a atualização e expansão das informações sobre produtos.

REFERÊNCIAS

[1] Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language* 13, 4 (1999), 359–394.
[2] Mateo Clement. 2025. The Role of AI Chatbots in Enhancing Customer Satisfaction in Service-Based Businesses. (2025).
[3] Aniket Jain and Sneh Soni. 2023. MarbleBot: A Conversational Recommender and Assistance Chatbot for Marble Selection Based on Dialogflow. In *2023 International Conference on Electrical, Electronics, Communication and Computers (ELEXCOM)*. IEEE, 1–6.
[4] Daniel Jurafsky and James H. Martin. 2023. *Speech and Language Processing* (3 ed.). Prentice Hall. Draft available at https://web.stanford.edu/ jurafsky/slp3/.
[5] Emmanuel Ok. 2025. How AI Chatbots Influence Customer Loyalty in E-Commerce and Service Industries. (2025).
[6] Muhammad Rahulil, Yuni Yamasari, Ricky Eka Putra, I Made Suartana, and Anita Qoiriah. 2025. A Systematic Literature Review on Chatbot Development For Whatsapp: Programming Language, Method, And Utility. *Jurnal Serambi Engineering* 10, 3 (2025).
[7] Alexandre Silva, Carolina Antunes, Rolando Miragaia, Rogério Luís Costa, Fernando Silva, and José Carlos Bregieiro Ribeiro. 2025. Artificial Intelligence Applied to the Stone Manufacturing Industry: A Systematic Literature Review. *Available at SSRN 5188439* (2025).