

---

Curso de Sistemas de Informação  
Universidade Estadual de Mato Grosso do Sul

---

**DESENVOLVIMENTO DE UM CHATBOT PARA DESTINOS  
TURÍSTICOS INTELIGENTES**

Patricia Nogueira Lima

Dr. Cleber Valgas Gomes Mira (Orientador)

Dourados-MS  
2024

# **DESENVOLVIMENTO DE UM CHATBOT PARA DESTINOS TURÍSTICOS INTELIGENTES**

Patricia Nogueira Lima

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Patricia Nogueira Lima e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Sistemas de Informação.

Dourados, 04 de novembro de 2024

Prof. Dr. Cleber Valgas Gomes Mira (Orientador)

---

L71d      Lima, Patrícia Nogueira

Desenvolvimento de um chatbot para destinos turísticos inteligentes / Patrícia Nogueira Lima. – Dourados, MS: UEMS, 2024.

57 p. : il.

Monografia (Graduação) – Sistemas de Informação – Universidade Estadual de Mato Grosso do Sul, 2024.

Orientador: Prof. Dr. Cleber Valgas Gomes Mira.

1. Chatbot 2. Turismo inteligente 3. Inteligência Artificial (IA) 4.  
Processamento de Linguagem Natural (PNL) I. Mira, Cleber Valgas Gomes II.

Título

CDD 23. ed. - 006.3

---

Curso de Sistemas de Informação  
Universidade Estadual de Mato Grosso do Sul

---

## **DESENVOLVIMENTO DE UM CHATBOT PARA DESTINOS TURÍSTICOS INTELIGENTES**

**Patricia Nogueira Lima**

Novembro de 2024

Banca Examinadora:

Prof. Dr. Cleber Valgas Gomes Mira (Orientador)

Área de Computação – UEMS

Prof. Dr. Diogo Fernando Trevisan

Área de Computação – UEMS

Prof. Dr. Rubens Barbosa Filho

Área de Computação – UEMS

## **AGRADECIMENTOS**

Gostaria de dedicar esta dissertação a todas as pessoas que de alguma forma contribuíram para o seu sucesso, com um agradecimento especial aos meus pais, que sempre me incentivaram a superar as dificuldades, à minha irmã e aos meus amigos, por seu constante apoio. Não posso deixar de mencionar meu orientador, cuja orientação e apoio foram fundamentais para que este trabalho fosse concluído com êxito. Quero também enfatizar a importância da perseverança diante das adversidades. Em momentos difíceis, é essencial não desistir e seguir em frente, pois é assim que alcançamos nossos objetivos.

*“Um computador mereceria ser chamado de inteligente se pudesse enganar um humano  
fazendo-o acreditar que era humano.”*

*( Alan Turing)*

## RESUMO

O avanço constante da tecnologia da informação e a crescente demanda por serviços de turismo personalizados têm impulsionado o desenvolvimento de soluções inovadoras no setor. Neste trabalho, apresentamos o projeto e a implementação de um *Chatbot* para Turismo Inteligente, com foco no estado de Mato Grosso do Sul. Esta aplicação é baseada em Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN) e tem como objetivo melhorar a experiência dos turistas, facilitando a interação com os destinos turísticos.

O *Chatbot* para Turismo Inteligente oferece aos turistas a capacidade de acessar informações relevantes sobre destinos, por meio de uma interface de conversação amigável. Utilizando técnicas avançadas de PLN, o *Chatbot* é capaz de compreender as consultas dos turistas e fornecer respostas precisas.

Este trabalho também enfatiza o papel fundamental desempenhado pela IA e pelo PLN na criação de soluções para o turismo inteligente. Em um cenário em que a personalização e a acessibilidade são essenciais para os viajantes, o *Chatbot* para Turismo Inteligente representa uma inovação promissora. Esperamos que este projeto contribua para a disseminação e a acessibilidade do turismo no estado de Mato Grosso do Sul, com a perspectiva de poder ser adaptado para outras regiões do Brasil no futuro.

**Palavras-chave:** Chatbot, turismo inteligente, IA, PLN, tecnologia.

# ABSTRACT

The constant advancement of information technology and the growing demand for personalized tourism services have been driving the development of innovative solutions in the sector. In this work, we present the design and implementation of a Smart Tourism Chatbot, focusing on the state of Mato Grosso do Sul. This application is based on Artificial Intelligence (AI) and Natural Language Processing (NLP) and aims to enhance the tourist experience by facilitating interaction with tourist destinations.

The Smart Tourism Chatbot offers tourists the ability to access relevant information about destinations through a using friendly conversational interface. Using advanced NLP techniques, the Chatbot can understand tourists' queries and provide accurate responses.

This work also highlights the essential role played by AI and NLP in creating solutions for smart tourism. In a scenario where personalization and accessibility are crucial for travelers, the Smart Tourism Chatbot represents a promising innovation. We hope that this project contributes to the dissemination and accessibility of tourism in the state of Mato Grosso do Sul, with the prospect of being adapted for other regions in Brasil in the future.

**Keywords:** Chatbot, smart tourism, AI, NLP, technology.

## LISTA DE FIGURAS

Figura1	Exemplo de um <i>chatbot</i> no site oficial de turismo do Japão. O <i>chatbot</i> responde uma pergunta sobre os sistemas de transporte do país. Fonte: < <a href="https://www.japan.travel/en/au/">https://www.japan.travel/en/au/</a> > . . . . .	13
Figura2	A página inicial do <i>site</i> de Copenhague. Fonte: < <a href="https://www.visitcopenhagen.com/">https://www.visitcopenhagen.com/</a> > . . . . .	22
Figura3	Um exemplo de interação no terminal do programa ELIZA. Fonte: < <a href="https://oldplayers.com.br/wp/2020/10/07/eliza-sendo-portado-para-varios-computadores/">https://oldplayers.com.br/wp/2020/10/07/eliza-sendo-portado-para-varios-computadores/</a> > . . . . .	24
Figura4	A página inicial do site . . . . .	41
Figura5	Seção destinadas as principais atrações do Estado de Mato Grosso do Sul . . . . .	42
Figura6	Chatbot respondendo a uma pergunta feita pelo usuário. . . . .	42

## **LISTA DE TABELAS**

Tabela1	Tabela de Perguntas e Respostas . . . . .	32
---------	---	----

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Objetivos . . . . .	14
1.2	Objetivos específicos . . . . .	14
1.3	Justificativa . . . . .	14
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>16</b>
2.1	Contexto do problema . . . . .	16
2.2	Destinos turísticos inteligentes e hospitalidade . . . . .	20
2.3	Processamento de Linguagem Natural (PLN) e Chatbots . . . . .	23
<b>3</b>	<b>Bibliotecas e <i>Frameworks</i></b>	<b>27</b>
3.1	Prolog . . . . .	27
3.2	Python . . . . .	28
3.3	Django . . . . .	28
3.4	PySwip . . . . .	29
<b>4</b>	<b>Desenvolvimento</b>	<b>30</b>
4.1	Funcionalidades . . . . .	30
4.1.1	Banco de Dados de Perguntas e Respostas . . . . .	30
4.1.2	Tratamento de Texto . . . . .	34
	Conversão para minúsculas . . . . .	35
	Remoção de pontuação . . . . .	35
	Substituição de palavras compostas . . . . .	35
	Tokenização . . . . .	35
	Remoção de <i>stopwords</i> . . . . .	35

Distribuição Peso . . . . .	35
Resultado Final do Processamento . . . . .	35
Pergunta Armazenada no Banco de Dados . . . . .	36
Resposta . . . . .	37
4.1.3 Cálculo de similaridade . . . . .	38
4.2 Interface Web . . . . .	41
<b>5 Conclusão</b>	<b>43</b>
.1 Apêndice A - Código Fonte do Chatbot Prolog . . . . .	44
<b>Referências Bibliográficas</b>	<b>52</b>

# 1 INTRODUÇÃO

No cenário atual, a inteligência artificial (IA) assumiu um lugar de destaque em nossas vidas. O que antes parecia ficção científica e algo reservado apenas para filmes agora se tornou uma parte integral do nosso cotidiano, ela desempenha um papel crescentemente crucial em uma variedade de setores econômicos, seja por meio de aplicativos ou *websites* (LEE, 2019).

A inteligência artificial tem demonstrado um potencial significativo em diversas áreas, e a indústria do turismo não é exceção. Mesmo que em um estado considerado principiante no seu desenvolvimento, a introdução dessa tecnologia pode representar uma revolução substancial na indústria de viagens e turismo. Isso se reflete não apenas na criação de empregos, mas também na melhoria de operações e na qualidade dos serviços prestados (ALMEIDA, 2019).

Diante desse cenário, os *Chatbots* e o Turismo Inteligente se destacam como soluções promissoras para aprimorar a comunicação entre os turistas e os lugares que visitam. Os *Chatbots* são como assistentes virtuais que podem conversar com as pessoas. Eles reconhecem as entradas dos usuários e utilizam correspondência de padrões para acessar informações e fornecer um reconhecimento predefinido (DAHIYA, 2017), enquanto o Turismo Inteligente usa a tecnologia para tornar a experiência turística mais interessante (BRANDÃO; JOIA; TELES, 2016). A integração dos *chatbots* no turismo inteligente surge com o propósito de aprimorar a experiência do turista e o bem-estar dos residentes locais (MEJÍA, 2022).

A implementação de um *chatbot* no setor de turismo enfrenta vários desafios significativos. Em primeiro lugar, é essencial que os *chatbots* respondam adequadamente às perguntas dos usuários, o que requer uma compreensão abrangente e precisa do *chatbot* para entender e responder às perguntas de forma eficaz. Além disso, o *chatbot* deve demonstrar habilidades sociais que estejam alinhadas com o modelo mental do usuário durante a conversa, tendo respostas de fácil compreensão para os turistas (CHAVES, 2023).

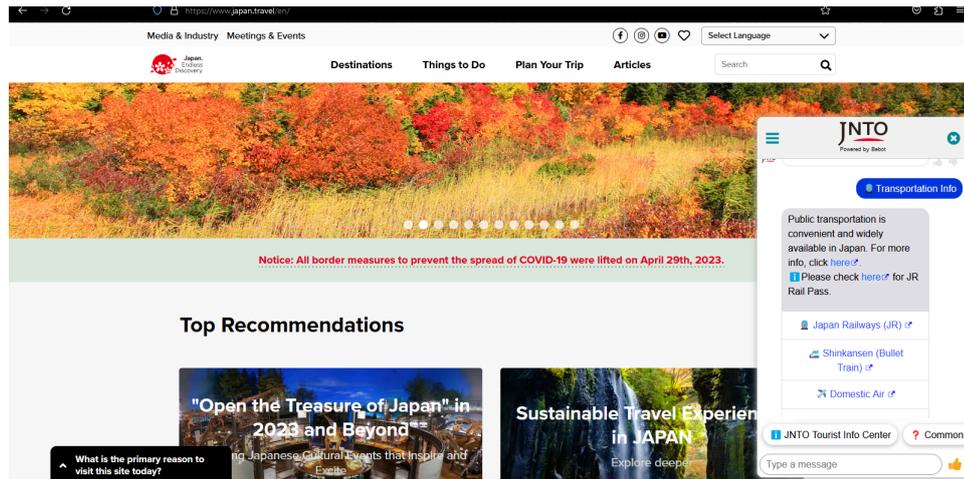


Figura 1: Exemplo de um *chatbot* no site oficial de turismo do Japão. O *chatbot* responde uma pergunta sobre os sistemas de transporte do país. Fonte: <<https://www.japan.travel/en/au/>>

No setor do turismo atualmente é mais comum encontrar *chatbots* que se concentram em cidades, regiões ou nichos específicos. Por exemplo, destinos como Japão (JAPAN, 2023) e Dubai (VISITDUBAI, 2023) têm *chatbots* dedicados em seus sites oficiais, que fornecem informações detalhadas de acordo com a perguntas individuais dos visitantes, como mostrado na Figura 1 onde, no exemplo apresentado, o *chatbot* responde à pergunta feita sobre meios de transporte do país. Com base no exemplo apresentado, torna-se evidente a necessidade de um *chatbot* integrado que possa responder a perguntas gerais sobre destinos e lugares, eliminando a exigência de visitar diversos *websites* para obter informações.

Este trabalho tem como objetivo apresentar o desenvolvimento e a implementação de um *Chatbot* para Turismo Inteligente focado em informações voltadas para o Estado de Mato Grosso do Sul, explorando a capacidade do Processamento de Linguagem Natural (PLN) para aprimorar a experiência do turista. Em resumo, este estudo busca contribuir para a compreensão do potencial dos *Chatbots* de Turismo Inteligente como ferramenta de melhoria da experiência dos turistas e no setor de turismo como um todo.

No Capítulo 1, é apresentada a introdução ao uso de *chatbots* como uma ferramenta que pode agregar valor ao setor de turismo. Contextualizamos exemplos de *chatbots* voltados para o turismo inteligente, apresentamos exemplos de *chatbot* e suas aplicações no turismo, bem como os objetivos deste trabalho e sua justificativa. No Capítulo 2, discutimos o turismo inteligente, as tecnologias de processamento de linguagem natural e *chatbots*. No Capítulo 3, apresentamos as bibliotecas e *frameworks* que serão usados no desenvolvimento deste trabalho.

No Capítulo 4, é apresentado todo o processo de desenvolvimento do trabalho, explicando como o *chatbot* foi implementado. Além disso, são descritas as principais funcionalidades do

*chatbot*, como o sistema de processamento de linguagem natural, que envolve a normalização de texto, a remoção de pontuação e a comparação de similaridade entre perguntas do usuário e o banco de dados de respostas. Também é discutida a maneira como a base de conhecimento foi organizada, incluindo a utilização de listas de palavras-chaves. O Capítulo 5 apresenta uma síntese final do trabalho analisando a relevância do desenvolvimento do *chatbot* baseado em *Prolog* para o setor de turismo. Serão discutidas as contribuições do projeto, como a viabilidade de utilizar técnicas de processamento de linguagem natural e similaridade ponderada para fornecer respostas precisas aos usuários. Além disso, o capítulo abordará as limitações identificadas e proporá direções para melhorias futuras.

## 1.1 Objetivos

Desenvolver um *chatbot* para turismo inteligente com o caso de estudo voltado para o estado de Mato Grosso do Sul, buscando melhorar a experiência do usuário na busca por informações, serviços turísticos.

## 1.2 Objetivos específicos

Os objetivos específicos são:

- Analisar as necessidades e demandas dos usuários do setor de turismo.
- Fazer um levantamento de tecnologias relacionadas a *chatbots*.
- Fazer um levantamento sobre tecnologias de inteligência artificial aplicada ao turismo.
- Desenvolver um protótipo de um *web site* de turismo com recursos de *chatbot*.
- Avaliar a usabilidade do *chatbot* de turismo inteligente.

## 1.3 Justificativa

Os *chatbots* têm ganhado cada vez mais popularidade em diversos setores, incluindo o turismo. Isso se deve ao fato de ser inviável o atendimento humano 24 horas por dia, onde os *chatbots* tornam possível essa interação a qualquer momento (LUGLI; FILHO, 2020).

Diante desse cenário, o trabalho propõe a utilização do *Prolog* em conjunto com técnicas de processamento de linguagem natural (PLN) para melhorar significativamente a qualidade

do *chatbot*. O treinamento do *chatbot* será baseado em informações relevantes sobre destinos turísticos, coletadas por meio de web sites e fóruns de turismo.

Utilizando o *Prolog* para inferência lógica e o PLN para o entendimento e manipulação da linguagem, o *chatbot* será capaz de fornecer respostas aos turistas, baseadas nas informações coletadas. O desenvolvimento de um *chatbot* inteligente para turismo nesse contexto representa uma oportunidade de explorar o potencial dessa tecnologia, combinando a lógica do *Prolog* com as capacidades avançadas de PLN para criar interações naturais e úteis para os usuários.

## 2 REVISÃO BIBLIOGRÁFICA

A Seção 2.1 apresenta o contexto do problema de utilização de *chatbots* para oferecer serviços em turismo inteligente. Serão apresentados estudos e pesquisas relacionados à implementação de tecnologias para destinos turísticos, além dos benefícios, desafios, melhores práticas e tendências emergentes no uso de *chatbots* para aprimorar a experiência do turista. A Seção 2.2 do capítulo aborda os conceitos de destinos turísticos inteligentes e hospitalidade, além de suas definições e características.

A Seção 2.3 explora o processamento de linguagem natural (PLN) e sua importância no desenvolvimento de *chatbots*. A seção apresenta uma visão geral do PLN, explicando como ele permite que máquinas compreendam e interajam com a linguagem humana. Também faz uma revisão histórica da evolução dos *chatbots*, desde os primeiros modelos baseados em regras até os mais modernos, que utilizam técnicas avançadas de PLN e aprendizado de máquina. Por fim, discute as principais técnicas de PLN utilizadas em *chatbots* e as tendências futuras para melhorar ainda mais a precisão e a naturalidade das interações.

### 2.1 Contexto do problema

A integração dos sistemas de informação, da internet e do setor do turismo impulsionou o surgimento de *websites* na *Web*, oferecendo aos usuários a conveniência de consultar, comprar e reservar produtos turísticos em apenas alguns cliques. Essa integração entre tecnologia e turismo proporciona aos viajantes uma experiência ágil e simplificada, permitindo que eles encontrem e adquiram facilmente os serviços desejados, seja hospedagem, passagens aéreas, pacotes turísticos ou atividades de lazer (RAMOS; RODRIGUES; PERNA, 2009a).

Quando os turistas decidem viajar, é natural buscar obter o máximo de informações disponíveis sobre o destino escolhido. Essas informações abrangem diversos aspectos, como opções de hospedagem, meios de transporte e atrações turísticas, além de informações específicas sobre o local, como acessibilidade para pessoas com deficiência. No entanto, ao chegarem no local escolhido, os turistas desejam obter mais informações, especialmente sobre como aproveitar ao

máximo sua estadia na cidade (ALVES et al., 2014).

Para garantir que as dúvidas e questionamentos dos turistas sejam atendidos de maneira rápida e eficaz, é essencial fornecer informações claras, precisas e atualizadas que permitam aos turistas explorar os principais pontos turísticos, conhecer a cultura local e desfrutar das atividades e eventos disponíveis (ALVES et al., 2014).

No setor de turismo, o uso da informação vai além da fase de planejamento e compra da viagem. Na verdade, a informação desempenha um papel fundamental nos serviços prestados no âmbito do turismo, desde um simples pedido em um restaurante típico até um *tour* pela cidade. É com base nesse princípio que as empresas do mercado turístico têm concentrado esforços em tecnologias de informação e comunicação, visando melhorar e tornar o produto do turismo mais seguro, padronizado e de melhor qualidade (PIRES, 2010).

Nesse contexto, surge a necessidade de adotar soluções tecnológicas inovadoras que possam aprimorar a experiência dos turistas. A análise de guias turísticos e a conversa com os agentes já não é a única opção para os viajantes planejarem suas viagens, uma vez que informações eletrônicas sobre viagens e as capacidades de reserva *online* estão cada vez mais disponíveis na *World Wide Web* onde os consumidores passaram a ter acesso direto a informações e serviços, sem a necessidade de recorrer a intermediários (ANA; GOMES, 2019).

Exemplos de serviços turísticos oferecidos em *websites*, portais e redes sociais são portais especializados, como *Booking* (BOOKING, 2023), *Airbnb* (AIRBNB, 2023), *Expedia* (EXPEDIA, 2023), entre outros. Esses serviços permitem aos usuários pesquisar diferentes opções de hospedagem, visualizar fotos e vídeos, ler avaliações de outros viajantes e realizar reservas online de forma conveniente. Além disso, os turistas podem utilizar *websites* como o *WikiTravel* (WIKITRAVEL, 2023) e o *TripAdvisor* (TRIPADVISOR, 2023) para obter mais informações sobre os destinos turísticos a partir de resenhas e avaliações coletadas por outros visitantes.

A hospitalidade esta diretamente ligada ao turismo, pois todo turista é recebido, de alguma maneira, nos lugares que visita. Todos os turistas de alguma forma ou outra acabam entrando em contato com alguém local, seja em um restaurante, seja num centro comercial, na rua ou um simples questionamento de uma orientação. A hospitalidade é um conceito que envolve a arte de receber e acolher visitantes de maneira calorosa e amigável (PLENTZ, 2005).

Essa capacidade proporciona uma experiência positiva e memorável aos turistas, garantindo que se sintam bem-vindos e cuidados durante sua estadia. No contexto do turismo, a hospitalidade vai além do simples atendimento ao cliente, envolvendo aspectos como a personalização do serviço, atenção aos detalhes e a criação de um ambiente acolhedor.

A hospitalidade é fundamental para estabelecer uma conexão emocional com os turistas, gerando satisfação e fidelização. Ela desempenha um papel crucial na formação da imagem de um destino turístico e na criação de experiências positivas que os turistas compartilham com outras pessoas. Portanto, investir em práticas de hospitalidade é essencial para o sucesso do setor turístico e para proporcionar aos visitantes uma estadia memorável e gratificante (OLIVEIRA; SANTOS, 2010).

Muitos *websites* no setor de turismo enfrentam desafios ao tentar atender plenamente os requisitos de hospitalidade e proporcionar uma experiência satisfatória aos usuários. Diversos fatores podem comprometer a eficácia na disseminação das informações, incluindo dificuldades de acesso a essas informações e falta de clareza nos textos. Além disso, muitos *sites* apresentam inconsistências na escolha de cores e na estrutura, o que pode resultar em erros e tornar a experiência desagradável para os usuários (GUARIZO; SCHWARTZ, 2020).

Com os avanços tecnológicos e a crescente conscientização sobre a importância da experiência do usuário, muitos *websites* de hotéis têm se esforçado para oferecer interfaces mais intuitivas, informações claras e recursos avançados. É possível encontrar *websites* de hotéis que oferecem ferramentas de reserva online eficientes, fotos e vídeos de alta qualidade das acomodações e ambientes, informações detalhadas sobre bares e restaurantes, políticas de hotel bem explicadas, e páginas iniciais atraentes e convidativas (OLIVEIRA, 2008).

Ainda assim, existem algumas dificuldades que o turismo enfrenta com relação aos *websites*. Uma das principais dificuldades é a sobrecarga de informações (COSTA, 2013).

A quantidade excessiva de informações em *sites* de turismo, pode dificultar a localização das informações relevantes pelos usuários. O conteúdo de *websites* para turismo deve abranger informações relevantes e essenciais, consideradas necessárias para que visitantes e turistas compreendam completamente o destino ou produto. A qualidade de um *site* é avaliada com base na presença de informações que sejam pertinentes aos usuários.

Isso engloba elementos como texto e recursos visuais no *site* do destino, como mapas interativos, informações úteis, como previsões de clima e horários de funcionamento de atrações, detalhes de contato, notícias, atrações adicionais, entre outros. Portanto, a organização adequada das informações é fundamental para proporcionar aos usuários uma experiência de busca por informações turísticas mais satisfatória (TONINI et al., 2023).

Um exemplo de falta de organização eficiente de informações em um *website* de turismo pode ser encontrado em um *site* que oferece guias turísticos para várias cidades ao redor do mundo. Suponha que o *site* tenha uma página principal repleta de *banners*, anúncios, *links* e menus de navegação confusos. Ao acessar a página, os usuários são bombardeados com

uma variedade de informações, sem uma estrutura clara ou destaque para as informações mais relevantes (AGNER, 2023).

Quando os usuários tentam localizar informações específicas, como atrações turísticas populares, horários de funcionamento, preços de ingressos ou orientações sobre como chegar a determinado local, eles podem se deparar com dificuldades. As informações podem estar espalhadas por várias páginas *web* ou enterradas em seções pouco visíveis, exigindo que os usuários naveguem por uma série de menus e *links* antes de encontrarem as informações às quais estão procurando (AGNER, 2023).

A navegação nos *sites* de turismo também pode ser complexa. A falta de uma estrutura de navegação clara e intuitiva, menus desorganizados e informações difíceis de encontrar podem frustrar os usuários e dificultar a busca por informações importantes (PERINOTTO; SOARES; SOARES, 2020).

Surge assim o conceito de turismo inteligente, que busca aproveitar as tecnologias digitais e a análise de dados para melhorar a experiência do turista, tornar os destinos mais sustentáveis e eficientes, e impulsionar o desenvolvimento econômico das comunidades locais (BRANDÃO; JOIA; TELES, 2016). O turismo inteligente envolve a aplicação de soluções tecnológicas inovadoras, como sistemas de informação, análise de dados, para criar uma infraestrutura turística mais inteligente e personalizada (BRANDÃO; JOIA; TELES, 2016).

Nesse sentido, os *chatbots* surgem como uma ferramenta promissora para o turismo, oferecendo a possibilidade de interação automatizada com os turistas, fornecendo informações, sugestões e auxílio no planejamento da viagem (ORDEN-MEJÍA; HUERTAS, 2022).

O objetivo deste trabalho é desenvolver um *chatbot* voltado para o estado de Mato Grosso do Sul que utilize tecnologias de modelos de linguagem natural para fornecer atendimento e assistência aos turistas, promovendo um ambiente de hospitalidade virtual. O *chatbot* será projetado para compreender as perguntas e solicitações dos usuários, oferecendo respostas relevantes e úteis. Além de fornecer informações sobre destinos turísticos, atrações e atividades, *chatbot* também irá ser capaz de auxiliar os turistas com informações sobre serviços.

Através da interação com o *chatbot*, os turistas poderão obter respostas rápidas para suas dúvidas comuns, como informações sobre transporte público, horários de funcionamento de atrações, recomendações de restaurantes e atividades culturais. A proposta é que o *chatbot* seja projetado com uma abordagem amigável e acolhedora, utilizando uma linguagem natural. Isso contribuirá para criar uma experiência de atendimento personalizada, simulando a interação com um guia turístico.

O objetivo final desse *chatbot* é melhorar a experiência dos turistas, oferecendo um suporte eficiente e conveniente durante suas viagens. Ao proporcionar um atendimento hospitaleiro, o *chatbot* poderá ser uma ferramenta valiosa para os turistas, fornecendo informações, auxiliando em suas necessidades e contribuindo para uma experiência de viagem mais satisfatória.

## 2.2 Destinos turísticos inteligentes e hospitalidade

Ao longo das últimas seis décadas, o turismo passou por um crescimento significativo e se tornou um dos setores econômicos mais relevantes e diversificados em escala global. Sua expansão acelerada trouxe consigo um impacto significativo no desenvolvimento socioeconômico de muitas regiões ao redor do mundo. O turismo deixou de ser apenas uma atividade de lazer e entretenimento, tornando-se uma indústria multifacetada que engloba diversas áreas, como hospitalidade, transporte, alimentação, cultura e lazer (ORGANIZATION, 2015).

No contexto do turismo, a tecnologia da informação e da comunicação (TIC) tem gerado valor para o setor e vem sendo utilizada intensamente pela demanda e, também, pela oferta turística, em diversas partes da cadeia produtiva. A aplicação do termo “inteligente” foi inserida no turismo, uma vez que, atualmente, existe um forte vínculo entre as TICs e o planejamento turístico. Nesse contexto, destaca-se o conceito de Destinos Turísticos Inteligentes (DTIs) (TA-ORI et al., 2023).

Atualmente, o termo “inteligente” tem sido amplamente utilizado para descrever o contexto de desenvolvimento tecnológico e inovação social e econômica. Ele está associado a conceitos como utilização de tecnologias avançadas, como *big data*, *open data* e sensores, que são fundamentais para o funcionamento da Internet das Coisas (*IoT*) (MAZO et al., 2021).

Além disso, a ideia de “inteligente” está relacionada à conexão, compartilhamento, integração, interação e sincronização em tempo real de informações, permitindo tomadas de decisão mais eficientes e assertivas. Essa abordagem busca aproveitar os recursos tecnológicos disponíveis para otimizar processos, melhorar a qualidade de vida das pessoas e impulsionar o progresso social e econômico (MAZO et al., 2021).

Um destino turístico é considerado inteligente quando possui uma série de elementos e requisitos essenciais. Para alcançar esse status, é necessário ter uma infraestrutura de conectividade eficiente, que garanta acesso à internet de alta velocidade e uma ampla cobertura de rede. Além disso, a utilização de tecnologias de informação e comunicação é fundamental, permitindo a integração de sistemas de informação, plataformas de reservas online e aplicativos móveis que facilitem o acesso a informações e serviços relevantes aos turistas (LIMA; FILHO,

2021).

A sustentabilidade também é um aspecto crucial, com a adoção de práticas ambientalmente conscientes e o fomento ao turismo responsável. A experiência do visitante deve ser personalizada e enriquecedora, com o uso de tecnologias interativas, sistemas de navegação e guias digitais. Por fim, a governança eficiente, com coordenação, políticas claras e gestão adequada, garante a qualidade dos serviços e a melhoria contínua do destino. Esses são os principais elementos e requisitos que contribuem para que um destino seja considerado inteligente, proporcionando uma experiência turística aprimorada e impulsionando o desenvolvimento sustentável (LIMA; FILHO, 2021).

A Comunidade Valenciana, na Espanha, é um exemplo de destino turístico inteligente que incorpora esses elementos em sua estratégia de turismo. Com o uso de tecnologias inovadoras, como *websites*, portais interativos e aplicativos móveis, a região oferece aos visitantes uma experiência aprimorada, permitindo-lhes explorar a riqueza cultural, as belezas naturais e as opções de lazer disponíveis na área (LIMA; FILHO, 2021).

Existem diversos exemplos de *websites* de destinos turísticos inteligentes que têm como objetivo oferecer uma experiência aprimorada aos visitantes. Um exemplo é Valencia, o *website* do destino turístico que fica na Espanha, que utiliza tecnologias avançadas para fornecer informações atualizadas sobre pontos turísticos, eventos, opções de transporte, hospedagem e atividades culturais (VISITVALENCIA, 2023).

Outro exemplo é o *website* de Copenhague, na Dinamarca, que oferece recursos interativos, como mapas digitais, roteiros personalizados e guias virtuais, para auxiliar os turistas a explorar a cidade de forma eficiente e personalizada (VISITCOPENHAGEM, 2023). Esses exemplos ilustram como os *websites* de destinos turísticos inteligentes estão se tornando uma ferramenta indispensável para os turistas na busca por informações e planejamento de viagens.

Os destinos turísticos inteligentes estão cada vez mais buscando aprimorar a experiência dos turistas e a hospitalidade desempenha um papel fundamental nesse contexto. A hospitalidade em destinos turísticos inteligentes envolve a aplicação de tecnologias e práticas inovadoras para proporcionar um acolhimento de qualidade aos visitantes (SOARES; CESARIO; ARAUJO, 2022).

Um exemplo no Estado de Mato Grosso do Sul é o município de Bonito foi selecionado para dois projetos que visam transformá-lo em um Destino Turístico Inteligente (DTI). No Programa Turismo Futuro Brasil, realizado em parceria com o Sebrae e o Banco Interamericano de Desenvolvimento (BID), Bonito está entre os 12 destinos brasileiros que recebem apoio para fortalecer suas estratégias e melhorar a competitividade e a qualidade turística. Além disso,

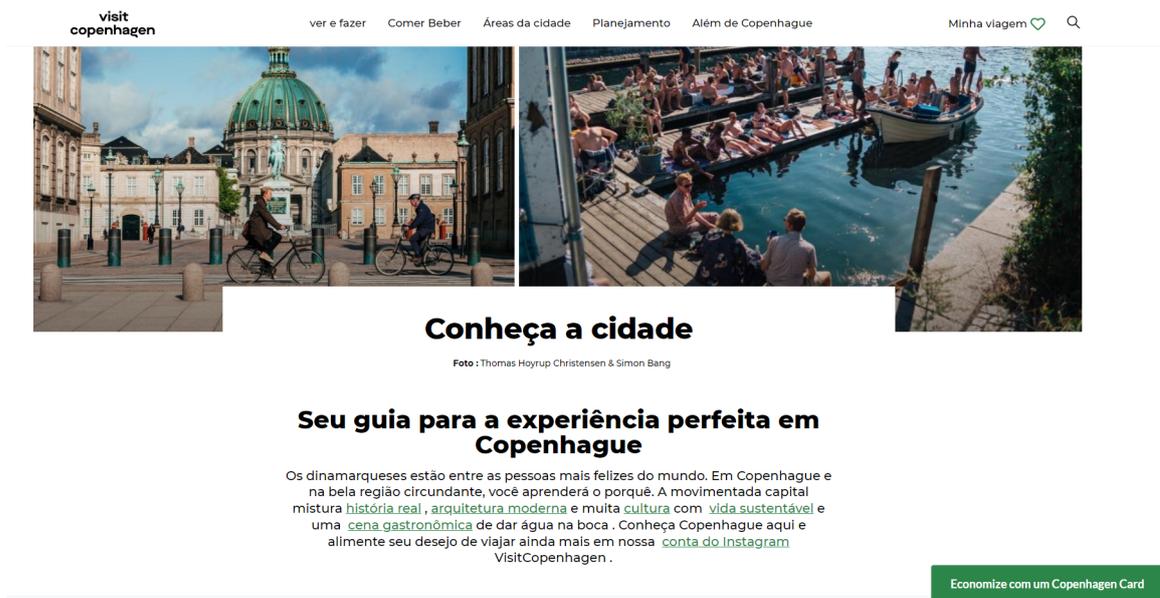


Figura 2: A página inicial do *site* de Copenhague. Fonte: <<https://www.visitcopenhagen.com/>>

o Projeto Estratégia Nacional Destinos Turísticos Inteligentes, em parceria com o Ministério do Turismo e o Instituto Cidades del Futuro, oferece suporte personalizado para a formulação e implementação de ações estratégicas. Ao final, Bonito receberá o certificado de "DTI em Transformação", sinalizando o progresso para se tornar um destino turístico inovador e inteligente (SUL, 2024) .

Além disso, a hospitalidade em destinos turísticos inteligentes pode envolver a disponibilização de acesso gratuito à internet em pontos turísticos e áreas públicas, permitindo que os visitantes fiquem conectados durante sua estadia. Essa conectividade facilita o acesso a informações úteis, como mapas *online*, avaliações de restaurantes e *feedback* de outros turistas, contribuindo para uma experiência mais conveniente e informada. A hospitalidade em destinos turísticos inteligentes também se estende à infraestrutura e serviços disponíveis. Esses destinos buscam oferecer instalações modernas e confortáveis, garantindo acesso fácil a serviços como hospedagem, transporte, alimentação e entretenimento (MAZO et al., 2021).

A TIC também tem contribuído para a comunicação e interação entre os turistas e os prestadores de serviços. Redes sociais, aplicativos móveis e *chats online* permitem que os turistas se conectem facilmente com os estabelecimentos e obtenham suporte e informações em tempo real. Isso melhora a comunicação, proporcionando respostas rápidas a dúvidas e necessidades dos turistas, aumentando a sensação de cuidado e atenção (FLORES; CAVALCANTE; RAYE, 2012).

A tecnologia da informação tem revolucionado a forma como a hospitalidade é oferecida nos destinos turísticos. Por exemplo, aplicativos móveis são amplamente utilizados para for-

necer informações detalhadas sobre pontos turísticos, restaurantes, eventos e atividades locais, permitindo que os turistas personalizem suas experiências de acordo com seus interesses e façam reservas com facilidade (RAMOS; RODRIGUES; PERNA, 2009b).

## 2.3 Processamento de Linguagem Natural (PLN) e Chatbots

O Processamento de Linguagem Natural (PLN) é uma área da inteligência artificial que se dedica ao estudo e desenvolvimento de algoritmos e técnicas para permitir que os computadores compreendam, interpretem e processem a linguagem humana de forma semelhante à capacidade dos seres humanos. O objetivo do PLN é capacitar os computadores a entender e responder a textos e comandos em linguagem natural, em vez de serem restritos apenas a comandos pré-definidos ou linguagens de programação específicas (CASELI; FREITAS; VIOLA, 2022).

A compreensão da linguagem natural tem sido um desafio significativo mesmo com os avanços nos últimos anos, ainda há problemas a serem resolvidos. Um dos principais desafios é a ambiguidade que uma palavra pode ter, ou seja, quando uma determinada palavra na frase possui mais de um sentido ou significado. Por exemplo, a palavra “banco” é ambígua, podendo ser entendida como uma instituição financeira ou como um assento.

Os seres humanos não têm muita dificuldade em lidar com essa ambiguidade, uma vez que possuem conhecimento cultural e experiência anterior para interpretar o contexto corretamente. No entanto, para as máquinas, essa tarefa é muito mais difícil e requer o desenvolvimento de técnicas avançadas do PLN e inteligência artificial para resolver a ambiguidade e realizar uma interpretação precisa do texto (DONIZETI, 2021).

Em 1950 Alan Turing propôs a pergunta “As máquinas podem pensar?”. Turing imaginou um “jogo de imitação”, hoje em dia conhecido como teste de Turing, no qual um humano “interrogador” faz perguntas por meio de um terminal a um outro humano e a uma máquina, sem saber previamente com quem está falando, com o objetivo de identificar quem é o humano e quem é a máquina (TURING, 1964).

Turing sugeriu que caso um computador enganasse o interrogador, ela demonstraria a capacidade de pensar, abrindo assim caminhos para a criação de algoritmos pioneiros. Um notável exemplo é a colaboração entre a *Georgetown University* e a IBM em 1954. Durante essa colaboração, um algoritmo inovador foi desenvolvido, demonstrando sua capacidade ao traduzir mais de sessenta sentenças do russo para o inglês. É importante ressaltar que esse feito ocorreu mesmo considerando que as informações eram inseridas por meio de cartões perfurados (GARVIN, 1968).

```

Welcome to

      EEEEE LL      IIII ZZZZZZ  AAAAA
      EE   LL      II   ZZ   AA  AA
      EEEEE LL      II   ZZZ  AAAAAA
      EE   LL      II   ZZ   AA  AA
      EEEEE LLLLL IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:   █

```

Figura 3: Um exemplo de interação no terminal do programa ELIZA. Fonte: <https://oldplayers.com.br/wp/2020/10/07/eliza-sendo-portado-para-varios-computadores/>

Um outro exemplo significativo foi o programa ELIZA, desenvolvido por Joseph Weizenbaum entre 1964 e 1966, tendo como abjetivo simular uma conversa com um psiquiatra, fazendo parecer que a pessoa estava conversando com um especialista, embora o ELIZA não fosse baseado em redes neurais, ele foi uma das primeiras tentativas de simular a conversa humana através de interações de texto (WEIZENBAUM, 1966). A Figura 3 mostra a execução do programa ELIZA em um terminal, onde uma pessoa faz perguntas e a ELIZA responde.

O processamento do ELIZA inicia o processo de resposta analisando a entrada do usuário em busca de palavras-chave específicas. Cada palavra-chave possui um *ranking* ou precedência atribuída pelo *script* ELIZA. Essas palavras-chave são colocadas em uma “pilha de chaves”, com a palavra de maior *ranking* no topo. Em seguida, a frase de entrada é manipulada e transformada de acordo com a regra associada à palavra-chave de maior *ranking*.

Por exemplo, no *script* DOCTOR, se palavras como “semelhante” ou “mesmo” forem detectadas na entrada, ELIZA geraria uma resposta relacionada à similaridade, como “De que maneira?”. Isso ocorre porque essas palavras têm um alto *ranking* de precedência. O *script* manipula essas palavras independentemente de considerações contextuais, como a troca de pronomes de primeira e segunda pessoa, já que o *ranking* de precedência é priorizado sobre os padrões convencionais de conversação.

O Prolog foi criado em 1973, na Universidade de Marseille, França. É uma linguagem

de programação lógica amplamente utilizada no desenvolvimento de sistemas de inteligência artificial, particularmente em sistemas de perguntas e respostas e *chatbots* (CLOCK SIN; MELLISH, 2003).

Sua estrutura baseada em regras lógicas e o foco em inferência o tornam adequado para a criação de aplicações que demandam raciocínio automatizado e processamento de linguagem natural. Ao contrário de outras linguagens, o Prolog permite definir um conjunto de fatos e regras, deixando que o sistema derive conclusões a partir dessas definições, o que facilita a implementação de sistemas que lidam com consultas e respostas (JÚNIOR et al., 2018).

Um exemplo clássico de aplicação do Prolog é na construção de *chatbots*, que podem interagir com os usuários por meio de perguntas e respostas automatizadas. O Prolog é eficaz para esse tipo de aplicação, pois permite o desenvolvimento de mecanismos de correspondência de padrões e dedução lógica que facilitam a compreensão de perguntas complexas (ESPINDOLA; SOUZA; CARVALHO, 2005).

Por exemplo, *chatbots* desenvolvidos em Prolog podem interpretar perguntas de usuários, comparar com um banco de dados de conhecimento e retornar a resposta mais relevante com base na similaridade entre as perguntas. Um exemplo desse tipo de aplicação é o *chatbot* desenvolvido para auxiliar o ensino de Espanhol como Língua Estrangeira, que utiliza o Prolog para processar perguntas dos alunos e oferecer respostas didáticas, facilitando o aprendizado interativo (FERREIRA, 2008).

Além disso, sistemas mais avançados podem realizar correções ortográficas e sugerir respostas com base em pesos atribuídos às palavras (ROSA; NETTO, 1994).

A década de 1980 marcou um grande avanço no campo do Processamento de Linguagem Natural (PLN) graças ao progresso significativo tanto em *hardware* quanto em *software*, com o uso de algoritmos de aprendizado de Máquina (ML). Agora os programadores não precisavam mais escrever manualmente todas as regras, pois os algoritmos de ML tinham a capacidade de aprender a partir de grandes conjuntos de dados (JOSEPH et al., 2016). A popularização das Redes Neurais Profundas (FRANÇA, 2021) na década de 2010, no entanto, foi o que causou um salto significativo na resolução de problemas de PLN. As Redes Neurais Profundas, especialmente as Redes Neurais Recorrentes (RNNs) e as Redes Neurais Convolucionais (CNNs), permitiram que modelos linguísticos mais complexos fossem treinados em grandes conjuntos de dados para tarefas como tradução automática, análise de sentimento e geração de texto (JOSEPH et al., 2016).

Finalmente, em 2019 foi desenvolvido o sistema *ChatGPT*. O *ChatGPT* é um exemplo avançado de um modelo de linguagem baseado em Redes Neurais Profundas, especificamente

a arquitetura *Transformer*. Essa tecnologia, desenvolvida pela *OpenAI*, tem a capacidade impressionante de compreender e gerar texto em linguagem natural de maneira coesa e próxima da linguagem humana. O *ChatGPT* é treinado em um enorme conjunto de dados de texto e usa técnicas de Aprendizado de Máquina para melhorar sua performance ao longo do tempo (OPENAI, 2021).

A inteligência artificial (IA) esta cada dia mais integrada ao nosso cotidiano e o *chatbot* é o exemplo mais proeminente dessa integração, representando uma das principais formas de interação entre máquinas e humanos. Um *chatbot* é um programa de computador que possui a capacidade de responder como uma entidade inteligente, que simula uma conversa humana por meio de texto ou voz. Essa tecnologia utiliza o processamento de linguagem natural para compreender as perguntas e declarações dos usuários e fornecer respostas adequadas e contextuais. Os *chatbots* também são referidos como *bots* inteligentes, agentes interativos, assistentes digitais ou entidades artificiais de conversação (ADAMOPOULOU; MOUSSIADES, 2020).

A interação entre um usuário humano e o *ChatGPT* ocorre em um terminal *web*, em um ambiente de um *site* ou uma plataforma, onde o usuário digita mensagens em uma caixa de texto. Essas mensagens podem ser perguntas, instruções ou comentários. Quando a mensagem é enviada, o *ChatGPT* processa o texto usando seu modelo de linguagem avançado. Com base na mensagem do usuário, o *ChatGPT* gera uma resposta. Essa resposta é exibida no terminal *web*, logo abaixo da mensagem original do usuário.

A conversa pode continuar com trocas adicionais de mensagens, permitindo que o usuário faça mais perguntas ou comente sobre a resposta recebida. A título de curiosidade, quando perguntado ao *ChatGPT* (OPENAI, 2021) sobre a definição de um *chatbot*, a sua resposta é: “Um *chatbot* é um programa de computador ou uma aplicação de inteligência artificial (IA) projetada para simular uma conversa com seres humanos por meio de texto ou voz, utilizando técnicas de PLN”.

## 3 BIBLIOTECAS E *FRAMEWORKS*

Este capítulo se dedica a apresentar e analisar as ferramentas tecnológicas essenciais que serão empregadas ao longo deste trabalho. Ao compreender o escopo e as capacidades dessas *frameworks* e bibliotecas, estaremos mais bem preparados para desenvolver um sistema de turismo inteligente. Portanto, este capítulo oferece uma visão geral das escolhas tecnológicas feitas.

### 3.1 Prolog

O Prolog, uma linguagem de programação declarativa, foi desenvolvido no início dos anos 1970 por Alain Colmerauer e Philippe Roussel, com o objetivo de manipular símbolos e realizar inferências lógicas de maneira eficiente. Amplamente utilizado em inteligência artificial, processamento de linguagem natural e sistemas especialistas, o Prolog destaca-se pelo paradigma declarativo, permitindo ao programador descrever o que deve ser alcançado (COZMAN; PLONSKI; NERI, 2021).

O conhecimento é representado por meio de fatos, que descrevem relações sempre verdadeiras, e regras, que possibilitam a inferência de novas informações a partir dos fatos existentes. O Prolog utiliza um mecanismo de resolução baseado em retrocessos (*backtracking*), que explora diferentes caminhos em busca de soluções que satisfaçam as consultas. Com sua capacidade de modelar conhecimento e executar inferências, o Prolog é uma ferramenta poderosa em áreas como sistemas especialistas, processamento de linguagem natural e inteligência artificial, onde o raciocínio lógico é essencial (CLOCKSIN; MELLISH, 2003).

No desenvolvimento do projeto, o Prolog foi utilizado tanto para a criação do banco de dados que armazena fatos e relações relevantes ao domínio do *chatbot*, quanto para o processamento das perguntas dos usuários. Essa abordagem permite ao sistema inferir respostas com base nas informações previamente cadastradas. A escolha do Prolog para o desenvolvimento do *chatbot* de turismo em Mato Grosso do Sul é apropriada devido à natureza limitada e específica do conteúdo abordado. Em um sistema com vocabulário restrito e pouca variabilidade, a abor-

dagem baseada em regras do Prolog se destaca, permitindo que o *chatbot* gerencie de forma eficiente as respostas a perguntas frequentes e específicas do contexto regional.

## 3.2 Python

A linguagem de programação *Python* é uma linguagem orientada a objetos, e interpretada, o que a torna uma escolha popular para programação de alto nível, com sua sintaxe simplificada e próxima da linguagem humana, tornando-a acessível para muitos programadores. Além disso, *Python* oferece uma vasta gama de *frameworks* e bibliotecas especializadas em Aprendizado de Máquina e Inteligência Artificial. *Python* ganha destaque como uma ferramenta essencial na análise de dados (LOPES et al., 2019). Sua utilidade se estende ao campo do “*web scraping*”, que envolve a automação de requisições a *sites*, análise de código *HTML*, extração de informações estruturadas e, em certas situações, até mesmo a interação com formulários (MITCHELL, 2019). Além disso, a versatilidade do *Python* se manifesta em áreas como desenvolvimento de jogos (SANTOS; MOREIRA, 2023) e em uma variedade de outros contextos, solidificando seu papel como uma linguagem de programação amplamente aplicável e flexível.

## 3.3 Django

*Django* é um *framework web* de alto nível, escrito em *Python*, que facilita o desenvolvimento rápido de aplicações seguras e escaláveis. Utilizando o padrão *Model-View-Template* (MVT), *Django* permite a criação de estruturas de dados robustas com seu poderoso sistema de mapeamento objeto-relacional (ORM), além de simplificar tarefas comuns, como autenticação e manipulação de formulários. Com uma interface administrativa pronta para uso, alto nível de segurança integrado e uma comunidade ativa, *Django* é amplamente utilizado em projetos de todos os tamanhos, desde pequenos *sites* até grandes plataformas como *Instagram* e *Pinterest* (SHAW et al., 2021).

Na implementação deste projeto, o *Django* foi utilizado como o *framework* principal para integrar a interface em *HTML* e *JavaScript* com o *backend* em Prolog, facilitando o processamento das consultas dos usuários, e gerenciando as requisições HTTP. Dessa forma, o *Django* atua como um intermediário, permitindo que as interações da interface em *HTML* e *JavaScript* sejam interpretadas pelo Prolog.

## 3.4 PySwip

O *PySwip* é uma interface que conecta o *Python* ao SWI-Prolog, permitindo que programas escritos em *Python* executem consultas em Prolog diretamente, sem a necessidade de interação manual com o ambiente Prolog. Essa biblioteca oferece funções para consultar, adicionar e remover fatos e regras da base de conhecimento Prolog, proporcionando uma integração perfeita entre os dois ambientes. A principal vantagem do *PySwip* é a possibilidade de usar a simplicidade e extensa biblioteca de *Python* para interface com o usuário, manipulação de dados e aprendizado de máquina, enquanto utiliza o Prolog para inferências lógicas complexas. Essa combinação permite desenvolver sistemas robustos e inteligentes, com a flexibilidade de *Python* e o poder lógico de Prolog (FOUNDATION, 2024).

## 4 DESENVOLVIMENTO

Neste capítulo, será abordado o desenvolvimento do *chatbot*, explicando a lógica implementada para o seu funcionamento e como essa implementação foi realizada. O *chatbot* foi desenvolvido utilizando a linguagem de programação Prolog, com o objetivo de responder perguntas relacionadas ao turismo. A arquitetura do sistema baseia-se em um banco de dados que contém perguntas e respostas pré-definidas, e em um processo de análise de similaridade que determina a melhor resposta com base na entrada do usuário.

### 4.1 Funcionalidades

#### 4.1.1 Banco de Dados de Perguntas e Respostas

O banco de dados de perguntas e respostas é uma funcionalidade essencial do *chatbot*, armazenando informações predefinidas que permitem ao sistema fornecer respostas relevantes aos usuários. Esse banco é constituído por fatos do tipo pergunta/resposta, onde cada fato contém um identificador, uma pergunta, uma resposta e uma lista de palavras chaves.

Essa estrutura facilita a busca por perguntas já catalogadas, identificando questões semelhantes àquelas feitas pelos usuários, o que aumenta a probabilidade de o *chatbot* forneça respostas adequadas e informativas (SHENOY; BHOOMIKA; ANNAIAH, 2022).

A coleta de perguntas e respostas foi realizada manualmente, abrangendo tópicos como hospedagem, restaurantes, passeios turísticos e eventos culturais. As respostas armazenadas incluem informações detalhadas, como nomes de restaurantes, atrações turísticas e eventos locais.

Os dados foram obtidos a partir de fóruns e sites de turismo, como Wikitravel (WIKITRAVEL, 2023) e Booking (BOOKING, 2023), utilizando palavras-chave como “turismo em Mato Grosso do Sul”, “restaurantes recomendados em Campo Grande”, “pontos turísticos em Bonito” e “eventos culturais em Corumbá”. A pesquisa também considerou redes sociais, como *Twitter* e *Reddit*, e fóruns como *TripAdvisor* e *Mochileiros*, onde foram analisadas as dúvidas mais frequentes dos turistas relacionadas ao estado de Mato Grosso do Sul.

Essas plataformas foram escolhidas devido à grande quantidade de interações e ao compartilhamento de experiências de viagens, permitindo identificar as principais preocupações e interesses dos visitantes na região. Essas informações foram então organizadas em um banco de dados na linguagem de programação Prolog, que serve como base de conhecimento para o *chatbot*. Por exemplo, considere o seguinte parágrafo do *Wikitravel* (WIKITRAVEL, 2023):

“O melhor período para se visitar Bonito é entre dezembro e março, na temporada do verão sul-matogrossense, período de chuvas, quando a vegetação está verde, os animais têm alimento abundante e os rios estão com o nível alto.”

A partir desse texto, uma pergunta gerada seria: “Qual é a melhor época do ano para visitar Bonito?”, com a resposta sendo:

“O melhor período para se visitar Bonito é entre dezembro e março, no verão sul-matogrossense, período de chuvas, quando a vegetação está verde, os animais têm alimento abundante e os rios estão com o nível alto. Entre maio e agosto, é a época de seca e as águas podem estar mais frias.”

Esse conjunto de pergunta e resposta é representado na base de conhecimento como um fato em Prolog:

`pergunta_resposta(1, ‘Qual é a melhor época do ano para visitar Bonito?’, ‘O melhor período para se visitar Bonito é entre dezembro e março, no verão sul-matogrossense, período de chuvas, quando a vegetação está verde, os animais têm alimento abundante e os rios estão com o nível alto. Entre maio e agosto, é a época de seca e as águas podem estar mais frias.’, [‘visitar’, ‘bonito’, ‘verão’, ‘maio’, ‘agosto’]).`

A Tabela 1 apresenta uma seleção de perguntas relacionadas às opções de entretenimento e passeios na cidade de Campo Grande. Cada entrada contém a identificação da pergunta, a questão formulada, a resposta correspondente e um conjunto de palavras-chave relevantes. Vale notar que as perguntas abordam o mesmo tema, sendo apresentadas em formas sinônimas.

No arquivo que contém as perguntas e respostas estão registradas um total de 3 mil perguntas, cada uma acompanhada de suas respectivas respostas e palavras-chave. Essas perguntas abordam diferentes aspectos das opções de entretenimento e atividades disponíveis em Mato Grosso do Sul.

O predicado **pergunta\_resposta** armazena um fato composto por um identificador, a pergunta, a resposta correspondente e uma lista de palavras-chave relacionadas à pergunta e resposta, e relevantes para o processamento de texto. Para melhorar a abrangência do *chatbot*, foram utilizadas ferramentas como o *ChatGPT* (OPENAI, 2021) para gerar frases sinônimas às

ID	Pergunta	Resposta	Palavras-Chave
10	Quais são as opções para participar de passeios ecológicos na região de Campo Grande?	Você pode participar de passeios ecológicos no Parque das Nações Indígenas e na Serra de Maracaju, conhecendo a fauna e flora local.	opções, participar, passeios ecológicos, região, Campo Grande, Parque das Nações Indígenas, Serra de Maracaju, fauna, flora local
11	Como são as opções de entretenimento noturno em Campo Grande?	A vida noturna em Campo Grande é animada, com diversas opções de bares, baladas e eventos culturais, especialmente nos fins de semana.	opções, entretenimento noturno, Campo Grande, vida noturna, animada, bares, baladas, eventos culturais, fins de semana
12	O que Campo Grande oferece para quem quer aproveitar a noite?	A vida noturna em Campo Grande é animada, com diversas opções de bares, baladas e eventos culturais, especialmente nos fins de semana.	Campo Grande, oferece, aproveitar a noite, vida noturna, animada, opções, bares, baladas, eventos culturais, fins de semana
13	Quais são os destaques da vida noturna em Campo Grande?	A vida noturna em Campo Grande é animada, com diversas opções de bares, baladas e eventos culturais, especialmente nos fins de semana.	destaques, vida noturna, Campo Grande, animada, opções, bares, baladas, eventos culturais, fins de semana

Tabela 1: Tabela de Perguntas e Respostas

já existentes, assegurando que diferentes variações na formulação das perguntas pudessem ser reconhecidas pelo sistema.

Isso foi necessário porque o número de perguntas geradas manualmente era limitado a cerca de 1.500 perguntas, o que poderia comprometer o funcionamento do *chatbot* e a experiência do usuário. Com um número maior de perguntas reescritas de formas variadas, o sistema se torna capaz de entender diferentes formulações da mesma questão, garantindo uma maior flexibilidade nas respostas e, conseqüentemente, uma experiência mais satisfatória para o usuário.

Uma abordagem adotada para otimizar o desempenho do *chatbot* foi a definição de listas específicas armazenadas em um banco de dados separado. Quando uma pergunta contém termos como “lista”, “restaurantes” e o nome de uma cidade, ao invés de realizar a busca nos fatos de **pergunta\_resposta**, o que poderia aumentar o tempo de processamento, a consulta é redirecionada para uma seção dedicada do programa.

Por exemplo, em vez de o *chatbot* realizar uma busca completa em todos os fatos armazena-

dos no banco de dados de perguntas e respostas (*pergunta\_resposta*), ele redireciona as consultas que contenham termos como “lista”, “restaurantes” e o nome de uma cidade para uma seção dedicada, que armazena apenas informações estruturadas sobre restaurantes, atrações turísticas ou hotéis disponíveis por cidade. Os dados sobre os restaurantes, hotéis e atrações turísticas foram obtidos do portal *TripAdvisor* (TRIPADVISOR, 2023).

No caso das consultas sobre restaurantes, os dados são armazenados de forma organizada, associando cada cidade a uma lista de estabelecimentos. Abaixo, um exemplo da estrutura utilizada no banco de dados:

```
restaurantes("Campo Grande", ["Famiglia Bressani Ristorante e Pizzeria", "Marruá Gourmet", "Yallah Restaurante"])
```

Quando o usuário faz uma pergunta que contém termos como “lista”, “restaurantes” ou variações, juntamente com o nome de uma cidade específica, o sistema não busca diretamente no banco de dados principal. Em vez disso, o *chatbot* utiliza um mecanismo de verificação que identifica esses termos e redireciona a consulta para a base de dados de restaurantes:

```
% Fato que armazena a lista de cidades
```

```
idades(['Campo Grande', 'Jardim', 'Bonito', 'Dourados'])
```

```
% Fatos de restaurantes por cidade
```

```
restaurantes('Campo Grande', ["Famiglia Bressani Ristorante e Pizzeria", "Marruá Gourmet", "Yallah Restaurante"])
```

```
verificar_pergunta(Pergunta, Restaurantes) :-
```

```
    atomics_to_string(Palavras, ' ', Pergunta),
```

```
    cidades(Lista_Cidades),
```

```
    intersecao(Palavras, Lista_Cidades, [Cidade|_]),
```

```
    (member('lista', Palavras);
```

```
    member('restaurante', Palavras);
```

```
    member('restaurantes', Palavras)),
```

```
    restaurantes(Cidade, Restaurantes).
```

A regra *verificar\_pergunta* processa a pergunta do usuário, dividindo-a em uma lista de palavras e verificando se alguma delas corresponde a termos como “lista” ou “restaurantes”. Em seguida, verifica se a pergunta inclui o nome de uma cidade presente no banco de dados. Se es-

sas condições forem atendidas, o *chatbot* acessa a base de dados e retorna a lista de restaurantes correspondente à cidade mencionada.

Para facilitar essa operação, uma regra auxiliar chamada *intersecao/3* é utilizada. Ela verifica a interseção entre as palavras da pergunta e a lista de cidades disponíveis, identificando corretamente o local mencionado pelo usuário:

```
intersecao([], _, []).
intersecao([H|T], L, [H|R]) :- member(H, L), intersecao(T, L, R).
intersecao([_|T], L, R) :- intersecao(T, L, R).
```

A implementação do banco de dados no *chatbot* desempenha um papel fundamental na eficiência e na precisão das respostas fornecidas. A estruturação das perguntas e respostas, juntamente com suas palavras-chave, permite uma busca otimizada, aumentando a probabilidade de geração de respostas corretas. A segmentação de informações em listas dedicadas, como restaurantes e atrações turísticas, aprimora ainda mais a performance do sistema, garantindo que consultas específicas sejam respondidas rapidamente.

No entanto, como a qualidade do sistema depende diretamente da completude do banco de dados e da variação nas perguntas armazenadas, é importante que esse conjunto seja periodicamente atualizado com novos dados e variações para continuar oferecendo boas respostas aos usuários.

### 4.1.2 Tratamento de Texto

O tratamento de texto é uma etapa crucial no processamento de linguagem natural (NLP). Ele engloba diversas técnicas para garantir que o texto seja processado de forma eficaz, preparando os dados textuais para que o *chatbot* possa analisá-los adequadamente (CASELI; NUNES, 2023).

Entre as técnicas mais comuns, está a remoção de *stopwords*, que são palavras frequentes, como artigos e preposições, que têm pouca relevância semântica. Paralelamente, o foco é dado às palavras-chave das perguntas, já que essas são essenciais para determinar a intenção do usuário.

Outro passo importante no tratamento de texto é a conversão de letras maiúsculas para minúsculas, o que padroniza as palavras, independentemente de sua formatação original. Além disso, a remoção de pontuação simplifica o texto, eliminando caracteres que não contribuem

para a análise. Após essas etapas, ocorre a *tokenização*, que divide o texto em unidades menores, como palavras e expressões, facilitando sua manipulação pelo *chatbot*.

Palavras compostas são tratadas de maneira especial, para evitar erros de tokenização. Por exemplo, o nome da cidade “Campo Grande” é transformado em “campogrande”, garantindo que ela seja tratada como uma única unidade. Essas técnicas combinadas asseguram que o texto seja limpo e padronizado, permitindo uma análise precisa e respostas adequadas por parte do sistema.

A seguir, apresentamos um exemplo prático de como o tratamento de texto ocorre a partir de uma pergunta digitada pelo usuário do *chatbot*.

**Conversão para minúsculas** Texto original: “Quais são as opções de hospedagem em Campo Grande?” Após conversão: “quais são as opções de hospedagem em campo grande?”.

**Remoção de pontuação** Texto original: “quais são as opções de hospedagem em campo grande?” Após remoção: “quais são as opções de hospedagem em campo grande”.

**Substituição de palavras compostas** Texto original: “quais são as opções de hospedagem em campo grande.” Após substituição: quais são as opções de hospedagem em campogrande”.

**Tokenização** Texto original: “quais são as opções de hospedagem em campogrande” Após tokenização: [“quais”, “são”, “as”, “opções”, “de”, “hospedagem”, “em”, “campogrande”].

**Remoção de stopwords** Texto original: [“quais”, “são”, “as”, “opções”, “de”, “hospedagem”, “em”, “campogrande”] Após remoção: [“são”, “opções”, “hospedagem”, “campogrande”, ]

**Distribuição Peso** Texto original:[“são”, “opções”, “hospedagem”, , “campogrande”, ] Após distribuição de pesos: [“são”: 1, “opções”: 1, “hospedagem”: 4,“campogrande”: 4]

**Resultado Final do Processamento** Lista de palavras-chave processadas:

- “são” (peso: 1),
- “opções” (peso: 1),

- “hospedagem” (peso: 4)
- “campogrande” (peso: 4).

*Stopwords* são palavras comuns em um idioma que não acrescentam informações significativas em muitos contextos, especialmente em tarefas de processamento de linguagem natural (PLN). Frequentemente, essas palavras são removidas de textos para melhorar a eficiência de algoritmos de análise, pois não contribuem para o significado ou para a distinção entre diferentes textos.

Exemplos de *stopwords* em português incluem artigos definidos e indefinidos como “o”, “a”, “os”, “as”, “um”, “uma”, “uns” e “umas”; preposições como “de”, “em”, “para”, “com”, “por”, “sobre”, “entre”, “até” e “sem”; conjunções como “e”, “ou”, “mas”, “porque”, “que” e “se”; pronomes como “eu”, “tu”, “ele”, “nós”, “vocês”, “eles”, “me”, “te”, “se”, “o”, “a”, “lhe” e “nos”; advérbios como “não”, “sim”, “também”, “sempre”, “já”, “muito” e “pouco”; além de outras palavras comuns como “que”, “isso”, “isto”, “aquilo”, “como”, “quando” e “onde”. A remoção dessas palavras ajuda a focar em termos mais relevantes durante a análise de texto.

No exemplo considerado, as palavras “são”: 1, “opções”: 1, “hospedagem”: 4, “campogrande”: 4 foram processadas com pesos específicos. Esses valores de peso que seguem uma escala de 1 a 4, do menos ao mais relevante, refletem a importância relativa de cada termo encontrado na pergunta e eles contribuem na formação da resposta do *chatbot*.

A utilização de pesos permite que o sistema realize um cálculo de similaridade ponderada, em que as palavras com maiores pesos exercem uma influência mais significativa na interpretação da consulta.

Essas palavras-chave serão comparadas com as palavras-chave das perguntas armazenadas na base de conhecimento, para determinar qual pergunta registrada é mais similar à pergunta do usuário. O banco de dados de perguntas e respostas contém palavras-chave específicas e relevantes para o setor de turismo, como nomes de cidades e termos turísticos comuns, como “passoio” e “lazer”. Cada lista de palavras-chave associada a uma pergunta no banco de dados ajuda a identificar a pergunta única correspondente.

Agora, vamos considerar uma pergunta já armazenada no banco de dados e ver como ela seria processada usando o procedimento descrito.

**Pergunta Armazenada no Banco de Dados** A pergunta armazenada no banco de dados é: “Onde encontrar uma hospedagem no centro de Campo Grande?”, com a resposta associada e as palavras-chave [“campogrande”, “hospedagem”, “centro”]. A inclusão da palavra “centro”

na lista de palavras-chave é fundamental, pois ela indica a localização da hospedagem. Isso é particularmente útil no caso em que o usuário menciona “hospedagem” na periferia de Campo Grande, permitindo ao *chatbot* distinguir entre diferentes áreas da cidade.

Para comparar as palavras-chave processadas da pergunta do usuário com as palavras-chave da pergunta armazenada, será utilizado o algoritmo de **similaridade do cosseno**. Detalhes sobre o cálculo da similaridade podem ser encontrados na Seção 4.1.3. Neste exemplo, a similaridade entre as duas listas de palavras-chave seria alta, uma vez que as palavras “hospedagem” e “campogrande” aparecem em ambas. Assim, o *chatbot* retornaria a seguinte resposta armazenada:

**Resposta** “Algumas opções de hospedagem no centro incluem Hotel Campo Grande e Grand Park Hotel.”

As palavras-chave cadastradas no banco de dados são escolhidas com base em sua relevância dentro do domínio do turismo. Cada conjunto de palavras-chave está associado a uma pergunta no banco de dados de perguntas e respostas e este conjunto será comparado com a lista de palavras geradas a partir da pergunta do usuário. Observe que, distintas perguntas podem estar associadas ao mesmo conjunto de palavras-chave, o que requer um processamento que considere os pesos das palavras-chave e também a sua ordem na sentença para identificar a pergunta/resposta mais adequada presente no banco de dados que corresponda à pergunta do usuário.

A similaridade entre as listas de palavras-chave é calculada com base no número de correspondências entre as palavras presentes na lista do usuário e nas listas das perguntas armazenadas. Além disso, cada palavra-chave recebe um peso específico, atribuído durante o processo de *tokenização*. Esse peso é utilizado para calcular a similaridade ponderada, aumentando a precisão da correspondência entre as perguntas do usuário e as perguntas armazenadas.

Por exemplo, suponha que, após o tratamento de uma pergunta feita pelo usuário, tenhamos a lista de palavras-chave: [“são”, “opções”, “hospedagem”, “campogrande”]. No banco de dados, existem várias listas de palavras-chave, como [“hospedagem”, “bonito”], [“hotel”, “campo grande”] e [“hospedagem”, “campogrande”, “centro”]. O processo de cálculo de similaridade identificará que a terceira lista é a mais semelhante à do usuário, permitindo que o *chatbot* identifique e retorne a resposta correta.

### 4.1.3 Cálculo de similaridade

O cálculo de similaridade é essencial para determinar a resposta mais adequada às perguntas dos usuários (RUBERTO; ANTONIAZZI, 2017). A similaridade cosseno ponderada (SCIENCE, 2022) é uma técnica que avalia a semelhança entre duas listas de palavras, levando em conta não apenas a presença ou ausência das palavras, mas também seus pesos específicos. Esses pesos são atribuídos conforme a importância das palavras em um determinado contexto, destacando termos relevantes para o domínio do problema. Por exemplo, em um *chatbot* médico, palavras como “sintoma”, “tratamento” e “doença” podem ter pesos mais altos em comparação com palavras como “dia” ou “de”.

Um dos principais desafios na aplicação de técnicas de similaridade é a escolha adequada dos pesos das palavras-chave. A atribuição desses pesos pode variar de acordo com o contexto e exige um entendimento profundo do domínio do problema (CAMPOS; FIGUEIREDO, 2021).

Durante o processamento de texto, as cadeias de caracteres são convertidas em duas listas: uma proveniente da pergunta do usuário e outra associada às perguntas já armazenadas no banco de dados. A similaridade entre essas listas de palavras-chave começa com a atribuição de pesos às palavras de acordo com sua relevância no domínio. Palavras comuns, que não contribuem diretamente para a identificação da pergunta ou que não pertencem ao domínio específico, recebem pesos menores (geralmente 1).

Por outro lado, palavras relevantes para o setor de turismo, como “hospedagem”, “restaurante” ou “passeio”, recebem pesos maiores, normalmente 4, dependendo de sua importância. Esses pesos refletem a relevância de cada termo para a identificação do tema da pergunta.

A atribuição de diferentes pesos às palavras-chave é uma estratégia fundamental para assegurar que, ao comparar a pergunta do usuário com as perguntas armazenadas no banco de dados, a ênfase seja colocada nas palavras mais relevantes. Por exemplo, se o usuário pergunta: “Quais são as opções de hospedagem em Campo Grande?”, e no banco de dados existe a pergunta “Hospedagem no centro de Campo Grande”, as palavras principais de cada uma são extraídas e recebem os pesos. No banco de dados, cada pergunta possui uma lista de palavras-chave associadas, que são armazenadas junto com um valor que reflete a relevância de cada uma dessas palavras para a consulta.

Por exemplo, para uma pergunta relacionada à “hospedagem”, “campogrande” e “centro” são armazenadas com um valor de peso 4, enquanto a palavra “centro” recebe o valor de peso 2. Esses valores são atribuídos com base na importância de cada termo para a interpretação e classificação da pergunta, sendo palavras mais relevantes aquelas que possuem maior peso.

Dessa forma, o cálculo de similaridade ponderada entre a pergunta do usuário e as perguntas armazenadas no banco pode priorizar as palavras-chave mais significativas, como “hospedagem” e “campogrande”.

A similaridade entre as perguntas é calculada com base nos pesos atribuídos a cada palavra-chave, garantindo que as palavras mais significativas tenham maior influência no cálculo. Assim, as palavras “hospedagem” e “campogrande”, com peso 4, possuem maior relevância na determinação da similaridade, enquanto “centro”, com peso 2, também contribui, mas de forma mais moderada. Dessa maneira, a resposta mais adequada ao usuário é definida com base nas palavras de maior peso.

Neste trabalho, as palavras-chave relevantes ao domínio do turismo foram identificadas e armazenadas em uma estrutura de dados específica para acesso rápido. Optou-se pela utilização de uma *Trie* (ou árvore de prefixos) (WIKIPEDIA, 2018), que permite armazenar as palavras-chave de forma eficiente, facilitando a busca e a comparação. A escolha da *Trie* é estratégica, pois ela possibilita que o sistema encontre rapidamente as palavras-chave relevantes armazenadas no banco de dados a partir da lista de palavras-chave recebidas como entrada do usuário.

Para aplicar a similaridade cosseno ponderada (SCIENCE, 2022) entre duas listas de palavras, cada palavra encontrada em uma das listas é representada como uma entrada em um vetor. Nesse vetor, cada elemento corresponde ao peso atribuído àquela palavra, refletindo sua relevância no contexto da pergunta. O processo de cálculo da similaridade cosseno pode ser descrito em detalhes a seguir.

Primeiramente, a representação de palavras como vetores é feita ao transformar cada lista de palavras em um vetor, onde a posição de cada palavra no vetor corresponde a uma dimensão específica. Por exemplo, para as palavras-chave “hospedagem”, “Campo Grande” e “centro”, o vetor resultante é o seguinte:

$$[\text{Vetor1} = [\text{peso}(\text{quais}), \text{peso}(\text{sao}), \text{peso}(\text{opcoes}), \text{peso}(\text{hospedagem}), \text{peso}(, ,) \text{peso}(\text{campogrande})]]$$

Se os pesos atribuídos a cada palavra forem 1, 2, 1 e 4, o vetor obtido é [1, 1, 1, 2, 1, 4]. Para o caso da lista de palavras contendo “hospedagem” e “Campo Grande”, nós obtemos o vetor

$$\text{Vetor1} = [\text{peso}(\text{hospedagem}), \text{peso}(\text{centro}), \text{peso}(\text{campogrande})] = [4, 2, 4]$$

Em seguida, a similaridade entre os dois vetores é calculada pela fórmula do cosseno, que mede o ângulo entre eles:

A similaridade do cosseno entre duas listas de palavras *A* e *B* é definida como:

$$\text{similaridade}(A, B) = \frac{|A \cap B|}{\sqrt{|A|} \times \sqrt{|B|}}$$

onde:

- $|A \cap B|$  é o número de palavras em comum (interseção) entre as listas  $A$  e  $B$ .
- $|A|$  é o tamanho da lista  $A$ .
- $|B|$  é o tamanho da lista  $B$ .

O valor da similaridade resultante varia entre 0 (nenhuma similaridade) e 1 (máxima similaridade), representando a relação entre as duas listas de palavras considerando os pesos definidos. Quanto menor o ângulo entre os vetores (ou maior o valor do cosseno), maior será a similaridade entre as listas de palavras. Por exemplo, dadas as listas [“hospedagem”, “campogrande”] e [“hospedagem”, “centro”, “campogrande”], a similaridade cosseno será calculada considerando os pesos atribuídos a cada palavra, ou seja, serão comparados os vetores [1, 1, 1, 2, 1, 4] e [4, 4, 2], neste exemplo o resultado é 0.7619047619047619.

O cálculo de similaridade é fundamental para determinar a resposta mais apropriada do *chatbot*. A similaridade cosseno ponderada permite que o sistema avalie a semelhança entre duas listas de palavras-chave, ponderando a relevância das palavras com seus respectivos pesos. Isso melhora a precisão na correspondência de uma resposta com a pergunta feita pelo usuário.

Um problema relacionado à similaridade entre perguntas é que, ao ignorar a ordem das palavras nas listas de palavras-chave, pode-se perder o significado original da pergunta. Isso ocorre porque o cálculo de similaridade utilizado considera apenas os pesos atribuídos às palavras, sem levar em conta a sequência em que elas aparecem. Em situações onde a ordem das palavras é fundamental para o sentido da frase, essa simplificação pode causar problemas. Por exemplo, as frases “O lobo correu atrás do coelho” e “O coelho correu atrás do lobo” contêm as mesmas palavras, mas em ordens diferentes, e por isso possuem significados opostos. No entanto, ao serem processadas como listas de palavras, essas duas frases seriam tratadas como equivalentes no cálculo de similaridade.

Para tratar casos em que as perguntas possuem as mesmas palavras, mas seus significados mudam devido à ordem, seria necessário incluir uma métrica que leve a ordem da sequência em consideração. Infelizmente, devido à limitação de tempo, não implementamos o cálculo dessa métrica de similaridade em nosso *chatbot*. A solução adotada foi selecionar a primeira pergunta retornada pela similaridade de cosseno e utilizá-la como base para a resposta, garantindo que o *chatbot* entregue uma resposta rápida e funcional.



Figura 4: A página inicial do site

Apesar dessas limitações, os experimentos preliminares de uso do *chatbot* indicam que a utilização da similaridade cosseno por si só já é capaz de tratar perguntas semelhantes de maneira consistente, oferecendo respostas razoáveis e de forma eficiente. Embora não capture todas as nuances do significado que podem ser influenciadas pela ordem das palavras, essa abordagem proporciona um bom equilíbrio entre precisão e eficiência no contexto do nosso sistema.

Para trabalhos futuros, uma análise mais objetiva sobre a qualidade das respostas oferecidas pelo *chatbot* pode ser uma área promissora de investigação. Essa análise poderia envolver métricas de avaliação que considerem não apenas a precisão das respostas, mas também a satisfação do usuário e a relevância das informações apresentadas.

## 4.2 Interface Web

A interface do *chatbot* foi projetada para os usuários que buscam informações sobre turismo no Mato Grosso do Sul. A estrutura da página *HTML* é organizada em várias seções que facilitam a navegação e a interação. O sistema foi implementado utilizando *Django*, *HTML*, *JavaScript* e *Python*, e adaptado a partir de um *template* já existente. Essa abordagem permite uma personalização eficiente e uma experiência de usuário intuitiva.

O cabeçalho contém um menu de navegação com links para as principais seções do site, como “Início”, “Cidades”, “Sobre” e “Contato”. A Figura 4 mostra a página principal da aplicação *web* com o menu de navegação.

Na seção “Atividades para todos”, são destacadas as principais atrações turísticas do estado. Para cada cidade, é atribuída a atração como mostrado na imagem Figura 5.

A interface do *chatbot* é integrada por meio de um formulário interativo que permite ao usuário inserir perguntas diretamente, como ilustrado na Figura 6. O formulário possui um campo de texto onde os visitantes podem digitar suas perguntas e enviá-las ao pressionar um botão de envio. Isso oferece um meio direto para que os turistas consultem o *chatbot* sobre



Figura 5: Seção destinada às principais atrações do Estado de Mato Grosso do Sul



Figura 6: Chatbot respondendo a uma pergunta feita pelo usuário.

suas dúvidas. Após o envio de uma pergunta, mas antes de ser processada pelo *chatbot*, o texto digitado pelo usuário é verificado por um corretor ortográfico (PYSPELLCHECKER, 2024) implementado como um módulo da aplicação *web*. A pergunta com as palavras corrigidas pelo corretor ortográfico é enviada ao sistema Prolog e processada para gerar uma resposta do *chatbot* que será exibida em área designada abaixo do formulário, mostrada na Figura 6. Esta interface que processa as perguntas e respostas facilita a comunicação e permite aos usuários visualizarem rapidamente as informações solicitadas.

Em suma, a interface do *chatbot* foi desenvolvida com o objetivo de ser amigável, assegurando que os visitantes do site possam encontrar facilmente as informações que procuram sobre o turismo no Mato Grosso do Sul.

## 5 CONCLUSÃO

Este trabalho demonstrou a viabilidade de um *chatbot* baseado em Prolog para responder a perguntas relacionadas ao turismo, utilizando técnicas de processamento de linguagem natural e similaridade ponderada. Os resultados indicam que o sistema consegue fornecer respostas úteis e relevantes, mesmo diante de limitações em casos de perguntas semanticamente semelhantes ou na identificação de palavras-chave. A interação entre o usuário e o *chatbot* ocorre em um terminal web, onde as perguntas feitas pelo usuário são processadas pelo Prolog, permitindo que o modelo de linguagem do sistema forneça respostas dinâmicas e informativas.

A representação de palavras como vetores, onde cada palavra é atribuída a um peso específico, facilita a análise de similaridade. A similaridade entre vetores é calculada pela fórmula do cosseno, que mede o ângulo entre eles, fornecendo uma métrica eficaz para determinar a relação entre diferentes perguntas e respostas.

Futuras melhorias podem incluir a implementação de uma análise mais aprofundada da ordem das palavras e o aprimoramento do sistema de identificação de palavras-chave, o que pode levar a uma maior eficiência e precisão nas respostas do *chatbot*. Além disso, uma análise da qualidade das respostas geradas pelo sistema pode ser uma direção interessante para trabalhos futuros, contribuindo para o desenvolvimento contínuo e aprimorado de *chatbots* na área de turismo e em outros domínios.

Por exemplo, podemos utilizar bases de dados públicas de perguntas e respostas para avaliar os algoritmos implementados neste trabalho, garantindo que o sistema esteja sempre aprendendo e se adaptando. Além disso, é interessante considerar a possibilidade de adaptar o sistema para um totem de informação, expandindo sua aplicabilidade e alcance no atendimento ao público.

Para enriquecer ainda mais a experiência do usuário, o *chatbot* pode não apenas fornecer respostas, mas também entender o contexto das perguntas e oferecer sugestões relevantes.

## .1 Apêndice A - Código Fonte do Chatbot Prolog

```
:- discontiguous intersecao/3.
```

```
% Lista de palavras compostas e suas substituições
```

```
palavras_compostas([
    ["campo grande", "campogrande"],
    ["morada dos baís", "moradadosbaís"],
    ["parque das nações indígenas", "parquedasnaçõesindígenas"],
    ["mato grosso do sul", "matogrossodosul"],
    ["boca da onça", "bocadaonça"],
    ["trilha adventure", "trilhaadventure"],
    ["trilha discovery", "trilhadiscovery"],
    ["meia trilha", "meiatrilha"]
]).
```

```
% Substituir palavras compostas no texto
```

```
substituir_palavras_compostas(Texto, TextoSubstituido) :-
    palavras_compostas(PalavrasCompostas),
    foldl(substituir_palavra, PalavrasCompostas, Texto, TextoSubstituido).
```

```
substituir_palavra([Composta, Substituida], Texto, TextoSubstituido) :-
```

```
    ( sub_string(Texto, Before, _, After, Composta) ->
        sub_string(Texto, 0, Before, _, BeforeText),
        sub_string(Texto, _, After, 0, AfterText),
        atomic_list_concat([BeforeText, Substituida,
            AfterText], ' ', TextoSubstituido)
    );
    TextoSubstituido = Texto % Se não encontrar a palavra composta,
    %mantém o texto original
).
```

```
% Função para converter texto para minúsculas
```

```
to_lowercase(Text, LowerText) :-
```

```

string_lower(Text, LowerText).

% Função para remover pontuação de uma string
remove_punctuation(Text, CleanText) :-
    atom_chars(Text, CharList),
    exclude(is_punctuation, CharList, CleanCharList),
    atom_chars(CleanText, CleanCharList).

is_punctuation(Char) :-
    char_type(Char, punct).

% Função para tokenizar texto em palavras
tokenize(Text, Tokens) :-
    split_string(Text, " ", "", Tokens).

% Lista de stopwords (palavras a serem removidas na análise)
stopwords(["a", "o", "as", "os", "um", "uma", "de", "em", "para", "com",
"por", "que", "e", "ou", "mas", "se", "não", "sim", "também", "muito",
"pouco", "isso", "isto", "aquilo", "como", "quando", "onde", "ele",
"ela", "nós", "você", "vocês", "eles", "me", "te", "se"]).

remove_stopwords([], []).
remove_stopwords([Token|Rest], CleanTokens) :-
    stopwords(Stopwords),
    (member(Token, Stopwords) ->
        remove_stopwords(Rest, CleanTokens)
        % Caso seja uma stopword, simplesmente prossegue
    ;
        remove_stopwords(Rest, CleanRest), % Caso não seja,
        %adiciona o token na lista limpa
        CleanTokens = [Token|CleanRest] % Adiciona o token
        %atual na lista de tokens "limpos"
    ),
    writeln(['Tokens após remover stopwords:', CleanTokens]). % Debugging

% Função para pré-processar texto completo

```

```

preprocess_text(Text, CleanTokens) :-
    to_lowercase(Text, LowerText),
    remove_punctuation(LowerText, CleanText),
    substituir_palavras_compostas(CleanText, SubstitutedText),
    tokenize(SubstitutedText, Tokens),
    remove_stopwords(Tokens, CleanTokens).

% Lista de palavras com pesos específicos
palavras_pesos([["centro", 2],["hospedagem", 4],["bocadaonca",4],
["discovery",4],["trilha",3],["adventure",4],["bodoquena",4],
["matogrossodosul", 4 ],["ms",4], ["comida", 2], ["passeio", 2],
["cultura", 2], ["campogrande", 4], ["bonito", 4], ["pantanal", 4],
["jardim", 4],
["trilhaadventure",4],["trilhadiscovery",4],["meiatrilha",4]]).

% Função para obter o peso de uma palavra
peso_palavra(Palavra, Peso) :-
    palavras_pesos(PalavrasPesos),
    (member([Palavra, Peso], PalavrasPesos) ->
        true;
        Peso = 1).

% Cálculo da similaridade ponderada
weighted_cosine_similarity(List1, List2, Similarity) :-
    % Encontrar os pesos das palavras na primeira lista (usuário)
    findall(Peso, (member(Palavra, List1), Palavra \= "",
    peso_palavra(Palavra, Peso)), Pesos1),

    % Encontrar os pesos das palavras na segunda lista (banco de dados)
    findall(Peso, (member(Palavra, List2), Palavra \= "",
    peso_palavra(Palavra, Peso)), Pesos2),

    % Encontrar a interseção entre as duas listas (palavras em comum)
    intersection(List1, List2, Intersection),

    % Atribuir pesos às palavras da interseção

```

```

findall(Peso, (member(Palavra, Intersection),
Palavra \= "", peso_palavra(Palavra, Peso)), PesosIntersection),

% Somar os pesos das interseções e das duas listas
sum_list(PesosIntersection, PesoIntersection),
sum_list(Pesos1, Peso1),
sum_list(Pesos2, Peso2),

% Calcular a similaridade com base nos pesos atribuídos
(Peso1 > 0, Peso2 > 0 ->
    Similarity is 2 * (PesoIntersection / (Peso1 + Peso2));
    Similarity is 0).
sum_list(PesosIntersection, PesoIntersection),
sum_list(Pesos1, Peso1),
sum_list(Pesos2, Peso2),

% Calcular a similaridade com base nos pesos atribuídos
(Peso1 > 0, Peso2 > 0 ->
    Similarity is 2 * (PesoIntersection / (Peso1 + Peso2));
    Similarity is 0).

% Função de similaridade
similaridade(Palavras1, Palavras2, Similaridade) :-
    weighted_cosine_similarity(Palavras1, Palavras2, Similaridade).

% Fato que armazena a lista de cidades
cidades(['Campo Grande', 'Jardim', 'Bonito', 'Dourados']).

% Fatos de restaurantes por cidade
restaurantes('Campo Grande', ['Famiglia Bressani Ristorante e Pizzeria',
'Marruá Gourmet', 'Yallah Restaurante']).
restaurantes('Jardim', ['Restaurante Pantanal', 'Churrascaria do Parque']).
restaurantes('Bonito', ['Casa do João', 'Juanita Restaurante', 'Toca do Peixe']).

```

```

restaurantes('Dourados', ['Restaurante do Gaúcho', 'Restaurante Dallas']).
restaurantes('Dourados', ['Restaurante Kikão', 'Cantina Mato Grosso']).

```

```

% Função para verificar se a pergunta é sobre restaurantes
e listar os restaurantes da cidade

```

```

verificar_pergunta(Pergunta, Restaurantes) :-
    preprocess_text(Pergunta, Palavras), % Preprocessa a pergunta
    %para remover pontuações e stopwords
    writeln(['Palavras processadas da pergunta:', Palavras]), % Debugging
    cidades(Lista_Cidades),
    intersecao(Palavras, Lista_Cidades, [Cidade|_]), % Verifica
    %se uma das palavras é uma cidade
    (member('lista', Palavras);
     member('restaurante', Palavras);
     member('restaurantes', Palavras)), % Verifica se a
    %pergunta contém palavras-chave relacionadas a restaurantes
    restaurantes(Cidade, Restaurantes), % Retorna a lista de
    %restaurantes da cidade correspondente
    writeln(['Restaurantes encontrados:', Restaurantes]). % Debugging

```

```

% Função auxiliar para interseção de listas

```

```

intersecao([], _, []).
intersecao([H|T], Lista, [H|Intersecao]) :-
    member(H, Lista),
    intersecao(T, Lista, Intersecao).
intersecao([_|T], Lista, Intersecao) :-
    intersecao(T, Lista, Intersecao).

```

```

% Adicionando essa nova verificação à função principal
melhor_resposta(Pergunta, MelhorResposta) :-
    preprocess_text(Pergunta, PalavrasPergunta),
    % Preprocessa a pergunta feita pelo usuário

    writeln(['Palavras processadas da pergunta:', PalavrasPergunta]),
    % Debugging

    (verificar_pergunta(Pergunta, Restaurantes) -> % Se a
    %pergunta for sobre restaurantes
        atomic_list_concat(Restaurantes, ', ', Resposta),
        MelhorResposta = Resposta
    ;
    findall([Resposta, Similaridade], (
        pergunta_resposta(_, _, Resposta, PalavrasChaveBase),
        % A comparação será feita diretamente com as palavras-chave da base

        writeln(['Palavras-chave da pergunta base:', PalavrasChaveBase]),
        % Debugging

        similaridade(PalavrasPergunta, PalavrasChaveBase, Similaridade),
        writeln(['Similaridade calculada:', Similaridade]),
        writeln(['Resposta correspondente:', Resposta,
        'Similaridade:', Similaridade])
    ), Resultados),
    writeln(['Lista de resultados:', Resultados]), % Debugging
    (Resultados = [] ->
        MelhorResposta = "Não compreendi a pergunta.
        Por favor, tente novamente.";
        max_similaridade(Resultados, MelhorResposta),
        writeln(['Melhor resposta:', MelhorResposta]) % Debugging
    )
).

```

```

% Função auxiliar para encontrar a interseção de duas
%listas (palavras e cidades)
intersecao([], _, []).
intersecao([H|T], Lista, [H|Intersecao]) :-
    member(H, Lista),
    intersecao(T, Lista, Intersecao).
intersecao(_|T], Lista, Intersecao) :-
    intersecao(T, Lista, Intersecao).

% Função auxiliar para encontrar a resposta com maior similaridade
max_similaridade([], [_, 0]). % Retorna 0 se a lista estiver vazia
max_similaridade([[Resposta, Similaridade]], [Resposta, Similaridade]).
% Caso base: apenas um elemento na lista

max_similaridade([[Resposta, Similaridade]|Resto], MelhorResposta) :-
    max_similaridade(Resto, [MelhorRespostaTemp, SimilaridadeTemp]),
    (Similaridade > SimilaridadeTemp ->
        MelhorResposta = [Resposta, Similaridade];
        MelhorResposta = [MelhorRespostaTemp, SimilaridadeTemp]
    ).

% Verifica se a entrada é uma saudação
eh_saudacao(Entrada, Resposta) :-
    preprocess_text(Entrada, Palavras),
    atomic_list_concat(Palavras, ' ', EntradaNormalizada),
    saudacao(EntradaNormalizada, Resposta).

% Código de saudações
saudacao("oi", "Olá! Como posso ajudar você hoje?").
saudacao("olá", "Oi! Em que posso ajudar?").
saudacao("bom dia", "Bom dia! Como posso ser útil?").
saudacao("boa tarde", "Boa tarde! Em que posso ajudar?").

```

```
saudacao("boa noite", "Boa noite! Como posso ajudar?").
```

```
% Função principal para interação com o usuário
```

```
main :-
```

```
    writeln('Iniciando chatbot...'), % Debugging
    loop.
```

```
loop :-
```

```
    write('Digite sua pergunta (ou "sair" para sair): '),
    read_line_to_string(user_input, PerguntaInput),
    writeln(['Pergunta recebida:', PerguntaInput]), % Debugging
    (
        PerguntaInput = "sair" -> (
            writeln('Finalizando chatbot!'),
            nl,
            fail
        );
        (
            writeln('Checando se é saudação...'),
            (eh_saudacao(PerguntaInput, Resposta) ->
                writeln(['Saudação detectada:', Resposta]),
                writeln(Resposta)
            );
            writeln('Não é saudação, buscando melhor resposta...'),
            melhor_resposta(PerguntaInput, Resposta),
            writeln(['Melhor resposta encontrada:', Resposta])
        )
    ),
    nl,
    loop % Continua o loop
).
```

## REFERÊNCIAS BIBLIOGRÁFICAS

- ADAMOPOULOU, E.; MOUSIADES, L. An overview of chatbot technology. In: SPRINGER. *IFIP international conference on artificial intelligence applications and innovations*. [S.l.], 2020. p. 373–383.
- AGNER, L. *Ergodesign e arquitetura da informação: trabalhando com o usuário*. [S.l.]: Digitaliza Conteúdo, 2023.
- AIRBNB. *Airbnb*. 2023. Acessado em: 2023-07-30. Disponível em: <<https://www.airbnb.com.br/>>.
- ALMEIDA, M. d. M. A. Robots, inteligencia artificial y realidad virtual: una aproximación en el sector del turismo. *Cuadernos de turismo*, n. 44, p. 13–26, 2019.
- ALVES, A. A. B. et al. O uso da informação no setor turístico. *Múltiplos Olhares em Ciência da Informação*, v. 3, n. 2, june 2014. Disponível em: <<https://periodicos.ufmg.br/index.php/moci/article/view/17425>>.
- ANA, S.; GOMES, A. *Turismo Brasileiro 4.0: do analógico ao digital. A digitalização e a mudança na venda de “pacotes” de viagens nas operadoras de turismo do Brasil*. Tese (Doutorado), 2019.
- BOOKING. *Booking*. 2023. Acessado em: 2023-07-30. Disponível em: <<https://www.booking.com/index.pt-br.html>>.
- BRANDÃO, M.; JOIA, L. A.; TELES, A. Destino turístico inteligente: um caminho para transformação. *Anais do Seminário da ANPTUR*, 2016.
- CAMPOS, S. L. B.; FIGUEIREDO, J. M. de. Uso de técnicas de processamento de linguagem natural para identificação de similaridade de serviços públicos. In: SBC. *Anais do IX Workshop de Computação Aplicada em Governo Eletrônico*. [S.l.], 2021. p. 83–94.
- CASELI, H.; FREITAS, C.; VIOLA, R. Processamento de linguagem natural. *Sociedade Brasileira de Computação*, 2022.
- CASELI, H. d. M.; NUNES, M. d. G. V. Processamento de linguagem natural: conceitos, técnicas e aplicações em português. 2023.
- CHAVES, A. P. Desenho de linguagem de chatbots: influência da variação da linguagem na experiência do usuário com chatbot assistente de turismo. In: SBC. *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas Colaborativos*. [S.l.], 2023. p. 70–75.
- CLOCKSIN, W. F.; MELLISH, C. S. *Programming in PROLOG*. [S.l.]: Springer Science & Business Media, 2003.
- COSTA, A. d. F. Hospitalidade virtual: um estudo sobre indicadores de hospitalidade nos sites de hotéis do rio de janeiro-rj. Niterói, 2013.

COZMAN, F. G.; PLONSKI, G. A.; NERI, H. Inteligência artificial. 2021.

DAHIYA, M. A tool of conversation: Chatbot. *International journal of computer sciences and engineering*, v. 5, n. 5, p. 158–161, 2017.

DONIZETI, P. D. A. *Desafios em Processamento de Linguagem Natural*. 2021.

Url<https://ufabcdivulgaciencia.proec.ufabc.edu.br/2021/09/08/desafios-em-processamento-de-linguagem-natural-v-4-n-9-p-3-2021/>.

ESPINDOLA, D. C. S.; SOUZA, J. P.; CARVALHO, M. A. Chatterbot: um sistema de inferência para agentes de conversação utilizando lógica de predicados. *Trabalho de Conclusão de Curso (Graduação em Ciência da Computação)–Universidade José do Rosário Vellano, Alfenas–MG*, 2005.

EXPEDIA. *Expedia*. 2023. Acessado em: 2023-07-30. Disponível em: [<https://www.expedia.com.br/>](https://www.expedia.com.br/).

FERREIRA, L. P. Desenvolvimento de um chatbot para auxiliar o ensino de espanhol 2008.

FLORES, L. C. da S.; CAVALCANTE, L. de S.; RAYE, R. L. Marketing turístico: Estudo sobre o uso da tecnologia da informação e comunicação nas agências de viagens e turismo de balneário camboriú (sc, brasil). *Revista Brasileira de Pesquisa em Turismo*, v. 6, n. 3, p. 322–339, 2012.

FOUNDATION, P. S. *pyswip*. 2024. Disponível em: [<https://pypi.org/project/pyswip/>](https://pypi.org/project/pyswip/).

FRANÇA, C. C. Detectando deepfakes em vídeos: uma abordagem utilizando redes neurais convolucionais residuais. 2021.

GARVIN, P. L. *The Georgetown-IBM experiment of 1954: an evaluation in retrospect*. [S.l.]: Mouton, 1968.

GUARIZO, M. A.; SCHWARTZ, G. M. Usabilidade de sites nos contextos do lazer, turismo e hospitalidade: uma análise da produção científica. *LICERE-Revista do Programa de Pós-graduação Interdisciplinar em Estudos do Lazer*, v. 23, n. 2, p. 362–385, 2020.

JAPAN. *visitjapan*. 2023. Disponível em: [<https://www.japan.travel/en/>](https://www.japan.travel/en/).

JOSEPH, S. R. et al. Natural language processing: A review. *International Journal of Research in Engineering and Applied Sciences*, v. 6, n. 3, p. 207–210, 2016.

JÚNIOR, C. F. de C. et al. Chatbot: uma visão geral sobre aplicações inteligentes. *Revista Sítio Novo*, v. 2, n. 2, p. 68–84, 2018.

LEE, K.-F. *Inteligência artificial*. [S.l.]: Globo Livros, 2019.

LIMA, D.; FILHO, L. M. Modelo invat. tur: Indicadores relevantes para gestão de um destino turístico inteligente. *Cenário: Revista Interdisciplinar em Turismo e Território*, v. 9, n. 2, p. 150–170, 2021.

LOPES, G. R. et al. Introdução à análise exploratória de dados com python. *Minicursos ERCAS ENUCMPI*, v. 2019, p. 160–176, 2019.

- LUGLI, V. A.; FILHO, J. de L. O uso do chatbot para a excelência em atendimento. *Revista Interface Tecnológica*, v. 17, n. 1, p. 205–218, 2020.
- MAZO, A. et al. Análise bibliográfica e sistemática da literatura acadêmica sobre “cidades inteligentes”, “turismo” e “competitividade”. *Turismo: Visão e Ação*, SciELO Brasil, v. 23, p. 148–168, 2021.
- MEJÍA, M. Á. Ó. *Análisis de los sistemas de diálogo chatbot en destinos turísticos inteligentes*. Tese (Doutorado) — Universitat Rovira i Virgili, 2022.
- MITCHELL, R. *Web Scraping com Python: Coletando mais dados da web moderna*. [S.l.]: Novatec Editora, 2019.
- OLIVEIRA, A. C.; SANTOS, M. d. No panorama conceitual da hospitalidade, a presença de novos aportes teóricos. *Anais do VI seminário de pesquisa em turismo do Mercosul. Caxias do Sul: Universidade Caxias do Sul*, 2010.
- OLIVEIRA, A. P. d. S. *Avaliação da usabilidade de Websites de hotéis segundo as heurísticas de Nielsen: pesquisa exploratória em hotéis da Região Nordeste do Brasil*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2008.
- OPENAI. *GPT-3.5 "ChatGPT" Model*. 2021. <<https://openai.com>>.
- ORDEN-MEJÍA, M.; HUERTAS, A. Evaluación de los atributos de los chatbots que son más efectivos en la interacción con el turista: Estudio de caso del chatbot “victoria la malagueña”. *Cuadernos de Turismo*, n. 50, p. 119–142, 2022.
- ORGANIZATION, W. T. *UNWTO Tourism Highlights, 2015 Edition*. Madrid, Spain: UNWTO, 2015.
- PERINOTTO, A. R. C.; SOARES, D. B.; SOARES, J. R. R. Websites oficiais como promoção turística: um estudo comparativo entre Brasil, França e Holanda à luz do marketing digital. *Revista Turismo Estudos e Práticas-RTEP/UERN*, v. 9, n. 2, p. 1–28, 2020.
- PIRES, L. C. Análise dos impactos da tecnologia de informação e comunicação para o turismo. *artigo*, 2010.
- PLENTZ, R. S. O papel da hospitalidade na busca de um outro turismo. *Anais do III SEMINTUR*, 2005.
- PYSPELLCHECKER. 2024. Acessado em: 4 de outubro de 2024. Disponível em: <<https://pypi.org/project/pyspellchecker/>>.
- RAMOS, C.; RODRIGUES, P.; PERNA, F. Sistemas e tecnologias de informação no sector turístico. *Revista Turismo Desenvolvimento*, v. 12, 2009.
- RAMOS, C.; RODRIGUES, P.; PERNA, F. Sistemas e tecnologias de informação no setor turístico. *Universidade de Aveiro*, 2009.
- ROSA, J.; NETTO, M. Lógica e conexionismo em processamento de linguagem natural (“logic and connectionism in natural language processing”), sucesusp’94 proceedings-ii jornada usp-sucesu-sp de informática e telecomunicações. *June. São Paulo, SP, Brasil*, 1994.

RUBERTO, D. L. V. G.; ANTONIAZZI, R. L. Análise e comparação de algoritmos de similaridade e distância entre strings adaptados ao português brasileiro. In: *Anais da XIII Escola Regional de Banco de Dados*. Porto Alegre, RS, Brasil: SBC, 2017. ISSN 2595-413X. Disponível em: <<https://sol.sbc.org.br/index.php/erbd/article/view/3035>>.

SANTOS, M. D. D.; MOREIRA, J. P. Jogo da força: Parte gráfica do jogo da força desenvolvido em python. *REFAQI-REVISTA DE GESTÃO EDUCAÇÃO EE TECNOLOGIA*, v. 14, n. 1, 2023.

SCIENCE, T. D. *What is Cosine Similarity? How to Compare Text and Images in Python*. 2022. Accessed: 2024-08-20. Disponível em: <<https://towardsdatascience.com/what-is-cosine-similarity-how-to-compare-text-and-images-in-python-d2bb6e411ef0>>.

SHAW, B. et al. *Web Development with Django: Learn to build modern web applications with a Python-based framework*. [S.l.]: Packt Publishing Ltd, 2021.

SHENOY, R.; BHOOMIKA, M.; ANNAIAH, H. Design of chatbot using natural language processing. *Knowledge Engineering for Modern Information Systems*. De Gruyter, p. 60–79, 2022.

SOARES, R.; CESARIO, M. F. P.; ARAUJO, M. V. P. de. Destinos turísticos inteligentes e a experiência dos turistas:: Uma revisão sistemática da literatura em periódicos nacionais de turismo. *Revista Rosa dos Ventos-Turismo e Hospitalidade*, v. 14, n. 3, 2022.

SUL, G. do Estado de Mato Grosso do. *Bonito - MS Turismo*. 2024. Accessed: 2024-10-11. Disponível em: <<https://www.turismo.bonito.ms.gov.br/dti>>.

TAORI, R. et al. *Alpaca: A Strong, Replicable Instruction-Following Model*. 2023. Url<<https://crfm.stanford.edu/2023/03/13/alpaca.html>>.

TONINI, H. et al. Avaliação de websites de regiões de enoturismo: casos brasileiros e internacionais. *Revista Brasileira de Pesquisa em Turismo*, SciELO Brasil, v. 16, p. e–2684, 2023.

TRIPADVISOR. *Tripadvisor*. 2023. Acessado em: 2023-07-30. Disponível em: <<https://www.tripadvisor.com.br/>>.

TURING, A. M. Computer machinery and intelligence. *Minds and Machines*, Prentice-Hall Englewood Cliffs, NJ, 1964.

VISITCOPENHAGEM. *visitcopenhagen*. 2023. Disponível em: <<https://www.visitcopenhagen.com/>>.

VISITDUBAI. *visitdubai*. 2023. Disponível em: <<https://www.visitdubai.com/pt/plan-your-trip>>.

VISITVALENCIA. *visitvalenciana*. 2023. Disponível em: <<https://www.visitvalencia.com/en>>.

WEIZENBAUM, J. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, ACM New York, NY, USA, v. 9, n. 1, p. 36–45, 1966.

WIKIPEDIA. *Trie*. 2018. Accessed: 2024-10-05. Disponível em: <<https://pt.wikipedia.org/wiki/Trie>>.

WIKITRAVEL. *Wikitravel*. 2023. Acessado em: 2023-07-30. Disponível em: <[https://wikitravel.org/pt/Página\\_principal/](https://wikitravel.org/pt/Página_principal/)>.