



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



MarIA - Um chatbot inteligente para atendimento de vítimas de violência contra a mulher

Tereza Stephanny de Brito Félix

Orientador: Prof. Dr. Carlos Manuel Dias Viegas

Natal, RN, 9 de novembro de 2025

MarIA Um chatbot inteligente para atendimento de vítimas de violência contra a mulher

Tereza Stephanny de Brito Félix

Orientador: Prof.Dr.Carlos Manuel Dias Viegas

Trabalho de Conclusão de Curso de Graduação na modalidade Monografia, submetido como parte dos requisitos necessários para conclusão do curso de Engenharia de Computação pela Universidade Federal do Rio Grande do Norte (UFRN/CT).

Natal, RN, 9 de novembro de 2025

Divisão de Serviços Técnicos

Catalogação da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Félix, Tereza S de B.

MarIA Um chatbot inteligente para atendimento de vítimas de violência contra a mulher / Tereza Stephanny de Brito Félix Natal, RN, 9 de novembro de 2025

65 p.

Orientador: Carlos Manuel Dias Viegas

Monografia Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Graduação em Engenharia de Computação.

1. Redação técnica Monografia. 2. L^AT_EX. I. Viegas, Carlos M D. II. MarIA Um Chatbot inteligente para atendimento de vítimas de violência contra a mulher

RN/UF/BCZM

CDU XXX.YYY (YYY.0)

Resumo

A violência contra a mulher é um problema estrutural que demanda soluções tecnológicas para facilitar o acesso das vítimas aos serviços de emergência. Este trabalho apresenta o desenvolvimento de um chatbot para o Telegram, utilizando Modelos de Linguagem de Grande Escala (LLM - Large Language Model) e técnicas de Inteligência Artificial (IA), para classificação e atendimento automatizado de ocorrências de violência contra a mulher. A solução foi projetada para permitir que vítimas entrem em contato de maneira discreta e segura, utilizando mensagens de texto, áudio, imagens, vídeos e localização. O chatbot processa as informações recebidas, classifica a gravidade da ocorrência e direciona os dados para os órgãos competentes, agilizando o atendimento. Além da implementação do sistema, este estudo discute os desafios técnicos, éticos e jurídicos envolvidos na aplicação da IA para segurança pública, destacando os impactos sociais da tecnologia no combate à violência de gênero.

Palavras-chave: Chatbot; Inteligência Artificial; Violência contra a Mulher; Linguagem Natural; Atendimento Emergencial.

Abstract

Violence against women is a structural problem that requires technological solutions to facilitate victims' access to emergency services. This work presents the development of a Telegram chatbot, utilizing Large Language Models (LLMs) and Artificial Intelligence (AI) techniques for automated classification and response to incidents of violence against women. The solution was designed to allow victims to contact emergency services in a discreet and secure manner, using text messages, audio, images, videos, and location data. The chatbot processes the received information, classifies the severity of the incident, and forwards the data to the appropriate authorities, streamlining the response process. In addition to the system implementation, this study discusses the technical, ethical, and legal challenges involved in applying AI to public safety, highlighting the social impact of technology in combating gender-based violence.

Keywords: Chatbot; Artificial Intelligence; Violence Against Woman, Natural Language Processing; Emergency Response.

Sumário

| | |
|--|------------|
| Sumário | i |
| Lista de Figuras | iii |
| Lista de Siglas | v |
| 1 INTRODUÇÃO | 1 |
| 1.1 MOTIVAÇÃO | 2 |
| 1.2 OBJETIVOS | 3 |
| 1.2.1 Objetivos Específicos | 3 |
| 1.3 ESTRUTURA DO TRABALHO | 4 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 5 |
| 2.1 VIOLÊNCIA CONTRA A MULHER: ASPECTOS CONCEITUAIS E ESTATÍSTICOS | 5 |
| 2.1.1 Definições e Tipos de Violência | 5 |
| 2.1.2 Panorama da Violência contra a Mulher no Brasil e no Rio Grande do Norte | 6 |
| 2.2 CHATBOTS E ASSISTENTES VIRTUAIS | 6 |
| 2.2.1 Conceito e Evolução dos <i>Chatbots</i> | 6 |
| 2.2.2 Arquitetura | 7 |
| 2.2.3 Tecnologias Aplicadas | 7 |
| 2.2.4 Aplicações Sociais e Sensíveis | 8 |
| 2.3 INTELIGÊNCIA ARTIFICIAL E PROCESSAMENTO DE LINGUAGEM NATURAL | 9 |
| 2.3.1 Fundamentos de Inteligência Artificial aplicados a diálogo | 9 |
| 2.3.2 Processamento de Linguagem Natural (PLN) | 9 |
| 2.3.3 Desafios e cuidados no uso de IA em contextos sensíveis | 9 |
| 2.4 PRIVACIDADE, SEGURANÇA E ÉTICA NO ATENDIMENTO DIGITAL | 10 |
| 2.4.1 Privacidade de Dados e Anonimato | 10 |
| 2.4.2 Segurança da informação em sistemas sensíveis | 10 |
| 2.4.3 Questões éticas e limitações do atendimento automatizado | 10 |
| 2.5 SOLUÇÕES EXISTENTES E LACUNAS ATUAIS | 11 |
| 2.5.1 Aplicativos, <i>chatbots</i> e canais de denúncia já disponíveis | 11 |
| 2.5.2 Limitações tecnológicas e sociais observadas | 11 |
| 2.5.3 Oportunidades para inovação e acolhimento digital | 12 |

| | |
|--|-----------|
| 3 TRABALHOS RELACIONADOS | 13 |
| 4 PROBLEMA | 15 |
| 4.1 CONTEXTUALIZAÇÃO DO PROBLEMA | 15 |
| 4.2 LIMITAÇÕES DAS SOLUÇÕES EXISTENTES | 15 |
| 4.3 FORMULAÇÃO DO PROBLEMA DE PESQUISA | 15 |
| 4.4 HIPÓTESE | 16 |
| 5 IMPLEMENTAÇÃO | 17 |
| 5.1 VISÃO GERAL DO DESENVOLVIMENTO | 17 |
| 5.2 ARQUITETURA E ORGANIZAÇÃO DO PROJETO | 17 |
| 5.2.1 Comunicação via API REST | 19 |
| 5.2.2 Justificativa do Uso de Microsserviços com <i>Docker</i> | 20 |
| 5.2.3 Estrutura dos Dados da API | 21 |
| 5.3 SERVIÇOS E INFRAESTRUTURA | 23 |
| 5.4 FERRAMENTAS E TECNOLOGIAS | 23 |
| 5.5 IMPLEMENTAÇÃO E MÓDULOS | 24 |
| 5.5.1 Módulo de PLN (API) | 24 |
| 5.5.2 Módulo de Integração com o <i>Telegram (BOT)</i> | 27 |
| 5.5.3 Módulo de Banco de Dados | 30 |
| 5.5.4 Módulo de Dashboard | 30 |
| 5.6 INTEGRAÇÃO E TESTES | 31 |
| 6 RESULTADOS | 33 |
| 6.1 RESULTADOS FUNCIONAIS | 33 |
| 6.1.1 Fluxo de Interação com a Usuária | 33 |
| 6.1.2 Registro Multimodal da Ocorrência | 34 |
| 6.1.3 Fluxos Diferenciais | 37 |
| 6.1.4 Síntese dos Resultados | 38 |
| 6.2 DASHBOARDS E MONITORAMENTO | 38 |
| 6.3 LIMITAÇÕES E DIFICULDADES | 41 |
| 6.4 VALIDAÇÃO DOS OBJETIVOS ESPECÍFICOS | 41 |
| 7 CONCLUSÃO | 43 |
| 7.1 RESPOSTA AO PROBLEMA DE PESQUISA | 43 |
| 7.2 CONTRIBUIÇÕES | 43 |
| 7.3 LIMITAÇÕES | 44 |
| 7.4 TRABALHOS FUTUROS | 44 |
| 7.5 CONSIDERAÇÕES FINAIS | 44 |
| Referências bibliográficas | 45 |

Lista de Figuras

| | | |
|------|--|----|
| 1.1 | Diagrama Preliminar | 2 |
| 1.2 | Diagrama de Fluxo | 4 |
| 5.1 | Arquitetura do sistema MarIA. | 19 |
| 5.2 | Fluxograma de processamento de um relato pelos agentes de PLN. | 27 |
| 5.3 | Diagrama de entidade e relacionamento do banco de dados. | 30 |
| 6.1 | Mensagem inicial de acolhimento. | 34 |
| 6.2 | Fluxo informativo. | 34 |
| 6.3 | Registro via áudio com transcrição. | 35 |
| 6.4 | Registro via mensagem de texto. | 35 |
| 6.5 | Compartilhamento de localização. | 36 |
| 6.6 | Envio de imagem anexa. | 36 |
| 6.7 | Etapa de conclusão do registro da ocorrência. | 37 |
| 6.8 | Exemplo de um fluxo de interação diferencial, ativado por palavra-chave, mostrando o início, a continuidade e a conclusão do registro de emergência. | 38 |
| 6.9 | Tela principal do Dashboard Análise geoespacial e listagem de ocorrências. | 39 |
| 6.10 | Dashboard Análise estatística e temporal. | 40 |
| 6.11 | Dashboard Análise de padrões por dia da semana e coocorrência de crimes. | 41 |

Nomenclatura

API Application Programming Interface

GPT Generative Pre-Trained Transformer

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation

LGPD Lei Geral de Proteção de Dados

LLM Large Language Model

OCR Optical Character Recognition

PLN Processamento de Linguagem Natural

REST Representational State Transfer

SQL Structured Query Language

Capítulo 1

INTRODUÇÃO

Este projeto descreve o desenvolvimento de um *chatbot* inteligente no *Telegram*, que integra Modelos de Linguagem de Grande Escala (*Large Language Model LLM*). A ferramenta que tratamos neste trabalho foi projetada especificamente para oferecer acolhimento e facilitar o registro de ocorrências de violência doméstica contra a mulher, permitindo que as vítimas relatem casos com maior segurança, sigilo e facilidade, superando algumas barreiras iniciais no acesso à justiça.

Para a clareza da exposição, é fundamental estabelecer duas definições centrais a este trabalho. Primeiramente, entende-se por *chatbot* um programa de computador que simula a conversação humana para automatizar tarefas e fornecer informações. Em segundo lugar, a violência contra a mulher é compreendida como qualquer ato baseado no gênero que resulte em dano físico, sexual ou psicológico, representando uma grave violação dos direitos humanos.

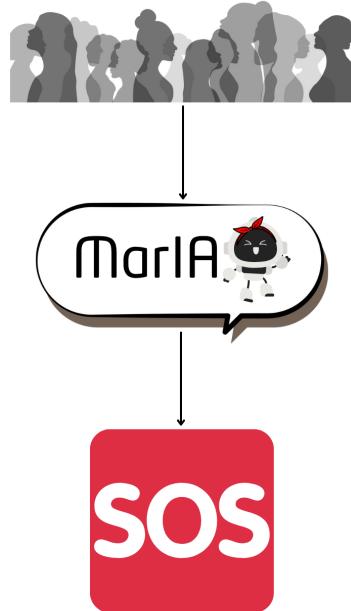


Figura 1.1: Diagrama Preliminar.

Fonte: Produção da Autora.

Além de servir como um canal de comunicação mais acessível para contatar os órgãos de segurança pública, conforme ilustrado na Figura 1.1, o sistema também poderá se tornar uma fonte de dados, os quais podem subsidiar a adaptação e o desenvolvimento de políticas públicas mais eficazes no combate à violência contra a mulher.

1.1 MOTIVAÇÃO

A violência doméstica é uma das mais graves violações aos direitos humanos no Brasil. De acordo com o Anuário Brasileiro de Segurança Pública (Fórum Brasileiro de Segurança Pública 2024), cresceram todas as modalidades de violência contra a mulher. E segundo o painel estatístico do Ministério da Justiça e Segurança Pública, durante o ano de 2024 foram registradas 1.510 mortes de mulheres em decorrência de feminicídios (Ministério da Justiça e Segurança Pública 2024). Apesar do aumento de campanhas de conscientização, muitas vítimas ainda enfrentam obstáculos para denunciar seus agressores entre eles, o medo, a vergonha, a dependência econômica ou emocional e a dificuldade de acesso aos meios de denúncia convencionais.

Esse cenário revela a urgência por soluções tecnológicas inovadoras, que ofereçam segurança, orientação e acolhimento às vítimas. O presente projeto propõe o uso de um *chatbot* inteligente baseado em Modelos de Linguagem de Grande Escala, como o GPT-4, capaz de receber informações, identificar e classificar automaticamente ocorrências de violência conforme os tipos previstos na Lei Maria da Penha (Brasil 2006).

1.2 OBJETIVOS

Este trabalho tem como objetivo desenvolver um sistema baseado em Inteligência Artificial (IA) para a captação, categorização e encaminhamento de ocorrências de violência doméstica contra a mulher, com foco no acolhimento inicial, na orientação das vítimas e na organização de dados úteis para órgãos públicos e entidades da sociedade civil.

A solução será implementada por meio de um *chatbot* acessível pelo aplicativo *Telegram*, visando garantir usabilidade e segurança para as usuárias. E contará com uma Interface de Programação de Aplicação (*Application Programming Interface API*) integrada a modelos de linguagem natural, responsável por interpretar as mensagens, classificar os relatos e encaminhá-los de forma adequada. Conforme ilustrado na Figura 1.2.

1.2.1 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos três objetivos específicos interligados. O primeiro consiste em desenvolver um *chatbot* integrado ao aplicativo *Telegram*, projetado para captar relatos de violência doméstica de forma segura e acessível, garantindo a privacidade das vítimas durante todo o processo de comunicação.

O segundo objetivo é integrar esse *chatbot* a uma API equipada com Inteligência Artificial, baseada em LLM. Essa integração tem a finalidade de interpretar e categorizar os relatos recebidos, além de oferecer orientações iniciais às vítimas e direcioná-las adequadamente aos órgãos competentes.

Por fim, o terceiro objetivo visa organizar e estruturar os dados obtidos de modo a torná-los acessíveis e úteis para órgãos de segurança pública, organizações não governamentais e instituições de apoio. Essa estruturação busca contribuir para a formulação e o aprimoramento de políticas públicas de enfrentamento à violência de gênero, em conformidade com a Lei Maria da Penha e demais diretrizes legais vigentes.

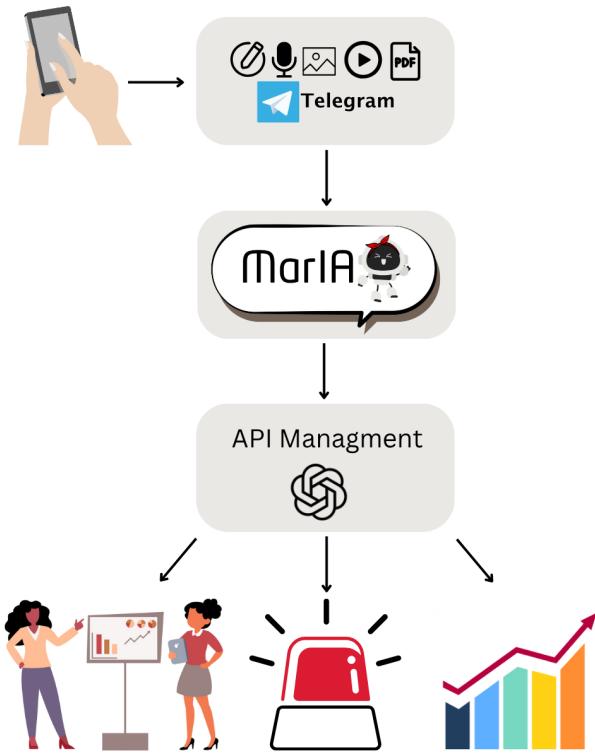


Figura 1.2: Diagrama de Fluxo.

Fonte: Produção da Autora.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está estruturado em capítulos, sendo o primeiro uma introdução sobre o tema, mostrando os fatores que motivam a implantação da ideia, além da justificativa e dos objetivos. Em sequência, o capítulo 2 aborda o referencial teórico. Os trabalhos relacionados e o problema são abordados nos capítulos 3 e 4 respectivamente. O capítulo 5, por sua vez, explica a metodologia para desenvolvimento do *chatbot* inteligente, enquanto o capítulo 6 trata dos resultados. Por fim, o capítulo 7 traz as principais conclusões e contribuições deste trabalho.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os fundamentos teóricos que sustentam o desenvolvimento deste trabalho. São abordados aspectos relacionados à violência contra a mulher, no contexto brasileiro e potiguar, incluindo dados estatísticos e legislações relevantes, com o intuito de contextualizar a importância de soluções tecnológicas voltadas ao acolhimento de vítimas. Em seguida, são discutidos os conceitos de *chatbots*, inteligência artificial e Processamento de Linguagem Natural (PLN), bem como questões éticas, de privacidade e segurança digital envolvidas no uso dessas tecnologias em contextos sensíveis. Por fim, examinam-se iniciativas similares já existentes, a fim de identificar lacunas e justificar a proposta deste projeto.

2.1 VIOLÊNCIA CONTRA A MULHER: ASPECTOS CONCEITUAIS E ESTATÍSTICOS

2.1.1 Definições e Tipos de Violência

A violência contra a mulher é um fenômeno complexo e multifacetado que atinge mulheres de diferentes idades, classes sociais, etnias e regiões. Segundo a lei brasileira número 11.340, De 7 de Agosto de 2006, conhecida como Lei Maria da Penha, a violência doméstica e familiar contra a mulher é qualquer ação ou omissão baseada no gênero que lhe cause morte, lesão, sofrimento físico, sexual ou psicológico, e dano moral ou patrimonial (Brasil 2006).

A lei também define cinco tipos principais de violência:

- **Violência física:** qualquer conduta que ofenda sua integridade ou saúde corporal;
- **Violência psicológica:** conduta que cause dano emocional, diminuição da autoestima, controle de comportamentos, ameaças, humilhações ou isolamento;
- **Violência sexual:** qualquer ação que a force a presenciar, manter ou participar de relação sexual não desejada;
- **Violência patrimonial:** retenção, subtração ou destruição de bens, documentos ou valores da mulher;
- **Violência moral:** calúnia, difamação ou injúria.

Essas formas de agressão se inter-relacionam e, frequentemente, coexistem no cotidiano das vítimas, com impactos profundos na saúde mental, na autonomia e no bem-estar das mulheres.

2.1.2 Panorama da Violência contra a Mulher no Brasil e no Rio Grande do Norte

Apesar dos avanços legais e institucionais, a violência contra a mulher permanece sendo um grave problema de saúde pública e de direitos humanos no Brasil. De acordo com (Fórum Brasileiro de Segurança Pública 2024), em 2023, o país registrou mais de 1.400 casos de feminicídio e mais de 245 mil casos de lesão corporal dolosa em contexto de violência doméstica. No estado do Rio Grande do Norte, a situação também é preocupante. Segundo levantamento do (DataSenado 2023), 29% das mulheres potiguaras afirmam já ter sofrido algum tipo de violência doméstica ou familiar, e 21% relataram violência nos doze meses anteriores à pesquisa. Dentre essas, 89% sofreram violência psicológica, 82% violência física, e 81% violência moral. Outro dado alarmante aponta que 40% das vítimas no estado sofreram a primeira agressão antes dos 19 anos de idade.

Além disso, houve um crescimento significativo na demanda por medidas protetivas. Em 2024, o estado registrou 13.005 medidas protetivas julgadas, um aumento de mais de 3 mil casos em relação a 2023 (Secretaria de Comunicação Social do Governo do Rio Grande do Norte 2025). Os atendimentos por meio da Central de Atendimento à Mulher — Ligue 180 — também cresceram, totalizando 10.276 atendimentos em 2024, com aumento de 13,7% em relação ao ano anterior, apesar de as denúncias formais terem apresentado uma leve queda.

Esses dados evidenciam a persistência da violência de gênero e a urgência de políticas públicas eficazes e acessíveis. A utilização de tecnologias digitais, como *chatbots*, pode representar uma estratégia complementar de apoio, acolhimento e orientação às vítimas, especialmente na superação de barreiras como o medo de denunciar, o isolamento ou a falta de acesso a canais formais.

2.2 CHATBOTS E ASSISTENTES VIRTUAIS

2.2.1 Conceito e Evolução dos *Chatbots*

Chatbots são sistemas computacionais capazes de interagir com seres humanos por meio da linguagem natural, geralmente por texto ou voz. Seu propósito é simular uma conversa humana de forma fluida, automatizando tarefas, respondendo perguntas ou guiando o usuário em fluxos predefinidos (Aquino & Adaniya 2018).

Os primeiros *chatbots* surgiram na década de 1960, como o famoso *ELIZA*, criado por Joseph Weizenbaum. Desde então, os avanços em processamento de linguagem natural, inteligência artificial e aprendizado de máquina (AM) tornaram os *Chatbots* muito mais sofisticados e eficientes (Jurafsky & Martin 2025).

Hoje, *Chatbots* são amplamente utilizados em serviços bancários, comércio eletrônico,

saúde e educação, desempenhando desde funções simples, baseadas em regras, até interações complexas com suporte a intenções, entidades e personalização do diálogo.

2.2.2 Arquitetura

A arquitetura de um *chatbot* pode variar conforme sua complexidade. *chatbots* simples são baseados em regras e funcionam por meio de árvores de decisão ou fluxos condicionais. Já os mais avançados utilizam técnicas de IA, especialmente PLN, para interpretar intenções, reconhecer entidades e gerar respostas mais naturais (Serban, Lowe, Henderson, Charlin & Pineau 2018).

A construção de chatbots modernos é frequentemente apoiada por *frameworks*, que podem ser definidos como um conjunto de ferramentas, bibliotecas e boas práticas que oferecem uma estrutura base para o desenvolvimento de software, agilizando o processo e promovendo a reutilização de código (Pressman & Maxim 2016). No ecossistema de agentes conversacionais, algumas das plataformas mais proeminentes incluem o *Dialogflow*, uma plataforma do *Google* que fornece ferramentas de PLN para projetar e integrar interfaces de conversação, destacando-se pela capacidade de reconhecer a intenção do usuário (Google 2024). Outra solução é o *Microsoft Bot Framework*, um serviço da *Microsoft* que oferece um ambiente de desenvolvimento integrado para a criação, teste e implantação de bots inteligentes em diversas plataformas (Microsoft 2024).

As tecnologias frequentemente empregadas incluem *frameworks* como *Dialogflow*, *Microsoft Bot Framework* e arquiteturas baseadas em modelos de linguagem como transformador generativo pré-treinando (*Generative Pretrained Transformer GPT*) (Brown, Mann, Ryder & et al. 2020). Esses sistemas combinam componentes como classificadores de intenções, extratores de entidades, geradores de resposta e mecanismos de diálogo.

Ademais, muitos desses *frameworks* integram funcionalidades como análise de sentimentos, contexto de sessão e conectores com bancos de dados e APIs externas, permitindo o desenvolvimento de *chatbots* capazes de fornecer serviços mais adaptados às necessidades dos usuários.

2.2.3 Tecnologias Aplicadas

Nesta seção, são apresentadas as principais tecnologias utilizadas no desenvolvimento da solução MarIA, incluindo o ambiente de execução *Node.js*, o sistema de banco de dados *PostgreSQL* e a plataforma de orquestração de contêineres *Docker*.

Node.js

JavaScript (JS) é, de acordo com o *Mozilla Developer Network* (MDN Web Docs 2022), uma linguagem de programação interpretada e orientada a objetos, que roda em muitos ambientes, sendo utilizada principalmente em navegadores.

O *Node.js* é um ambiente de execução de *JavaScript* do lado do servidor (*server-side*), de código aberto e multiplataforma. Ele permite que os desenvolvedores executem código *JavaScript* fora de um navegador web, sendo amplamente utilizado para construir aplicações de rede escaláveis, como APIs e microsserviços. Sua arquitetura orientada a

eventos e seu modelo de I/O (entrada e saída) não bloqueante o tornam eficiente para lidar com múltiplas conexões simultâneas (OpenJS Foundation 2023).

PostgreSQL

O *PostgreSQL* é um sistema de gerenciamento de banco de dados objeto-relacional de código aberto, conhecido por sua robustez, conformidade com os padrões linguagem de consulta estruturada (*Structured Query Language SQL*) e conjunto avançado de funcionalidades. Ele oferece suporte a uma vasta gama de tipos de dados, incluindo suporte nativo a *JSON*, e permite a criação de funções e procedimentos armazenados em diversas linguagens. Sua confiabilidade e escalabilidade o tornam uma escolha popular para aplicações que demandam alta integridade de dados (PostgreSQL Global Development Group 2024).

Docker

Docker é uma plataforma de código aberto que automatiza a implantação, o dimensionamento e a gestão de aplicações por meio da tecnologia de contêineres. Um contêiner encapsula uma aplicação e todas as suas dependências, bibliotecas, configurações e arquivos, em um ambiente isolado e portátil. Isso garante que a aplicação funcione de maneira consistente em diferentes ambientes, desde o desenvolvimento local até a produção em nuvem, simplificando a orquestração de arquiteturas baseadas em microsserviços (Docker, Inc. 2024).

2.2.4 Aplicações Sociais e Sensíveis

Nos últimos anos, os *chatbots* têm sido empregados em contextos mais delicados, como suporte emocional, saúde mental, acolhimento de vítimas de violência e mediação de conflitos. Nesses casos, a comunicação precisa ser cuidadosamente projetada para transmitir empatia, segurança e confiabilidade (Bibault, Chaix, Guillemassé & et al. 2019).

O uso de *chatbots* em ambientes sensíveis exige cuidados técnicos e éticos adicionais: é necessário garantir a proteção dos dados dos usuários, o anonimato, quando possível, a transparência das respostas e a limitação do escopo do atendimento. Esses fatores são essenciais para evitar o agravamento de situações de vulnerabilidade e criar um ambiente de acolhimento seguro.

Iniciativas como o *chatbot Aurora* (Brasil), criado para informar mulheres em situação de violência, ou o *Woebot* (EUA), focado em apoio psicológico, demonstram o potencial dessas ferramentas na promoção da saúde, segurança e cidadania digital (Ministério Público de São Paulo 2021, Woebot Labs Inc. 2017).

2.3 INTELIGÊNCIA ARTIFICIAL E PROCESSAMENTO DE LINGUAGEM NATURAL

2.3.1 Fundamentos de Inteligência Artificial aplicados a diálogo

Inteligência Artificial é um campo da ciência da computação voltado ao desenvolvimento de sistemas capazes de simular comportamentos inteligentes, como raciocínio, aprendizado e tomada de decisões. Quando aplicada ao contexto de diálogos com humanos, a IA é responsável por interpretar intenções, compreender a linguagem natural e gerar respostas coerentes (Russell & Norvig 2010).

A área de agentes conversacionais tem evoluído a partir de técnicas de aprendizado de máquina supervisionado e não supervisionado, redes neurais artificiais, aprendizado profundo (*deep learning*) e, mais recentemente, modelos de linguagem de larga escala, como o GPT (Brown et al. 2020). Esses modelos são treinados com grandes volumes de dados textuais e são capazes de prever a próxima palavra ou frase com base no contexto, o que os torna altamente eficientes para tarefas de geração de linguagem e conversação.

2.3.2 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural é uma subárea da IA voltada ao estudo e desenvolvimento de sistemas que possam compreender, interpretar e gerar a linguagem humana em forma textual ou falada (Jurafsky & Martin 2025). O PLN é essencial para o funcionamento de *chatbots*, pois permite a análise de mensagens do usuário, a identificação de intenções, a extração de entidades e a formulação de respostas.

Dentre as tarefas clássicas do PLN, destacam-se: análise morfossintática, reconhecimento de entidades nomeadas, análise de sentimentos, tradução automática e geração de texto. Em sistemas conversacionais, essas tarefas são aplicadas em tempo real para possibilitar interações dinâmicas e contextualizadas.

Diversos *frameworks* e bibliotecas têm contribuído para a popularização do PLN, como spaCy, NLTK, *Transformers* da *Hugging Face*, além das APIs disponibilizadas por grandes provedores de nuvem (*Google Cloud*, *Amazon Web Services*, *Azure*) para análise semântica, classificação de texto e detecção de intenção.

2.3.3 Desafios e cuidados no uso de IA em contextos sensíveis

Apesar de seu potencial, o uso de IA em contextos sensíveis — como o atendimento a vítimas de violência — exige uma abordagem responsável e cuidadosa. Modelos de linguagem podem reproduzir vieses presentes nos dados de treinamento, gerar respostas inadequadas ou mesmo induzir conclusões erradas (Bender, Gebru, McMillan-Major & Shmitchell 2021).

Além disso, a interpretação emocional de mensagens (como sinais de angústia, medo ou hesitação) ainda é um desafio para a IA, e muitas vezes requer intervenção humana em sistemas híbridos. A transparência dos algoritmos, a supervisão humana, o controle de

escopo e o respeito à privacidade do usuário são princípios fundamentais para a aplicação ética da IA nesses cenários (Jobin, Ienca & Vayena 2019).

Por isso, o desenvolvimento de soluções como o *chatbot* proposto neste trabalho deve considerar não apenas os avanços tecnológicos, mas também diretrizes éticas, jurídicas e sociais que assegurem o bem-estar e a proteção dos usuários em situação de vulnerabilidade.

2.4 PRIVACIDADE, SEGURANÇA E ÉTICA NO ATENDIMENTO DIGITAL

2.4.1 Privacidade de Dados e Anonimato

O atendimento a vítimas de violência exige o cumprimento de princípios fundamentais de privacidade e proteção de dados. Em muitos casos, a exposição da identidade da vítima pode representar risco real à sua integridade física e emocional. Por isso, qualquer sistema de atendimento automatizado deve garantir mecanismos de anonimato, controle de consentimento e proteção contra rastreamento digital (Brasil 2018).

No Brasil, a Lei Geral de Proteção de Dados (LGPD) estabelece regras claras sobre o tratamento de dados pessoais, exigindo base legal para coleta, armazenamento e compartilhamento de informações sensíveis. A aplicação dessas diretrizes é especialmente crítica em sistemas que interagem com populações vulneráveis, como mulheres em situação de violência.

2.4.2 Segurança da informação em sistemas sensíveis

A segurança da informação envolve a proteção de dados contra acessos não autorizados, perdas, vazamentos e manipulações. No caso de um *chatbot* voltado ao atendimento de vítimas, falhas de segurança podem comprometer a confiança no sistema e colocar em risco a vida das usuárias.

A adoção de protocolos de criptografia, autenticação segura, controle de acesso e auditoria de *logs* é essencial para minimizar riscos técnicos. Além disso, é importante considerar o uso de infraestrutura de hospedagem segura, bem como testes regulares de segurança para identificação de vulnerabilidades (Anderson 2020).

2.4.3 Questões éticas e limitações do atendimento automatizado

O uso de inteligência artificial em contextos sensíveis levanta diversas questões éticas. Dentre elas, destacam-se a transparência do sistema, o risco de respostas inadequadas, a ausência de empatia humana e a dificuldade em identificar situações de emergência ou perigo iminente (Burr, Cristianini & Ladyman 2020).

Embora os *chatbots* possam funcionar como ferramentas de acolhimento inicial, não devem substituir o atendimento humano qualificado, sobretudo em casos graves ou com complexidade emocional. É necessário que o sistema seja capaz de reconhecer seus

próprios limites e redirecionar a usuária, quando necessário, a canais de atendimento humano, como delegacias especializadas ou centros de apoio psicológico (Jobin et al. 2019).

A incorporação de princípios éticos desde a concepção do sistema — conhecida como *ethics by design* — é uma estratégia recomendada para mitigar impactos negativos e promover o uso responsável da tecnologia.

2.5 SOLUÇÕES EXISTENTES E LACUNAS ATUAIS

2.5.1 Aplicativos, *chatbots* e canais de denúncia já disponíveis

Nos últimos anos, diversos recursos digitais foram desenvolvidos para auxiliar mulheres em situação de violência. Dentre eles, destacam-se aplicativos como o PenhaS, desenvolvido pelo Instituto AzMina, que conecta vítimas a serviços de apoio, e o aplicativo Maria da Penha Virtual, iniciativa do Tribunal de Justiça do Rio de Janeiro, que permite o pedido de medida protetiva por meio de uma plataforma digital (Instituto AzMina 2024, Tribunal de Justiça do Estado do Rio de Janeiro 2021).

Além dessas iniciativas, observa-se um esforço governamental em âmbito estadual para criar canais digitais de apoio. O governo do Rio de Janeiro, por exemplo, lançou o aplicativo Rede Mulher, que centraliza serviços de atendimento (Governo do Estado do Rio de Janeiro 2024). Em São Paulo, o aplicativo SP Mulher oferece um botão de pânico e acesso rápido a delegacias (Governo do Estado de São Paulo 2023). No Paraná, foi implementado o Botão do Pânico virtual (Governo do Estado do Paraná 2024), e em Minas Gerais, o programa MG Mulher integra diversas ações de proteção (Governo do Estado de Minas Gerais 2024). Paralelamente, soluções da sociedade civil como o SOS Maria da Penha buscam oferecer agilidade no acionamento de ajuda (SOS Maria da Penha 2024).

Tais soluções representam avanços importantes na democratização do acesso à informação, sobretudo para mulheres em situação de vulnerabilidade social ou isolamento.

2.5.2 Limitações tecnológicas e sociais observadas

Apesar das iniciativas, há desafios significativos quanto à eficácia, acessibilidade e segurança desses sistemas. Muitos aplicativos têm alcance limitado, seja por barreiras tecnológicas (falta de internet, baixa familiaridade com interfaces digitais) ou institucionais (integração insuficiente com redes de apoio e órgãos públicos). Além disso, nem todos os sistemas garantem confidencialidade ou oferecem respostas realmente úteis em momentos críticos (Caldeira, Caldeira, Lima & Silva 2024).

Outro ponto crítico está relacionado à personalização e empatia do atendimento. Usuárias podem sentir-se frustradas ou não acolhidas por sistemas genéricos, que não interpretam adequadamente situações complexas. Em especial no caso de vítimas de violência, a percepção de acolhimento e a clareza das orientações são determinantes para a continuidade do uso do serviço (Jobin et al. 2019).

2.5.3 Oportunidades para inovação e acolhimento digital

Diante dessas limitações, surge a necessidade de soluções que integrem tecnologias de linguagem natural, inteligência artificial e princípios éticos desde a concepção. Um *chatbot* capaz de dialogar de forma sensível, oferecer informações contextualizadas e conduzir a usuária de forma segura a canais de ajuda pode ser um recurso complementar valioso.

Além disso, há oportunidades na integração com bancos de dados públicos e sistemas de denúncia automatizada, bem como na adaptação cultural e linguística das interfaces, de modo a atender à diversidade das usuárias. A construção de ferramentas tecnológicas sensíveis às especificidades da violência de gênero pode contribuir significativamente para o enfrentamento desse problema social.

Capítulo 3

TRABALHOS RELACIONADOS

Nos últimos anos, diversas soluções tecnológicas têm sido desenvolvidas com o objetivo de apoiar mulheres em situação de violência. Dentre essas iniciativas, destacam-se o aplicativo PenhaS, criado pelo Instituto AzMina, que conecta usuárias a serviços de apoio e fornece informações sobre seus direitos (Instituto AzMina 2024); o *Maria da Penha Virtual*, do Tribunal de Justiça do Estado do Rio de Janeiro, que permite o pedido online de medidas protetivas (Tribunal de Justiça do Estado do Rio de Janeiro 2021); e o aplicativo Salve Elas, desenvolvido por meio de parceria entre o Instituto Metrópole Digital (IMD/UFRN) e o Governo do Rio Grande do Norte, que permite que as mulheres com medida protetiva deferida realizem o acionamento discreto da Polícia Militar via integração com o Centro Integrado de Operações de Segurança Pública (CIOSP), indicando localização geográfica da usuária em situação de risco (Instituto Metrópole Digital / Governo do Rio Grande do Norte 2023).

Essas ferramentas representam avanços significativos na democratização da informação e no acesso à rede de proteção.

No entanto, estudos apontam que muitas dessas soluções ainda enfrentam limitações quanto à empatia no atendimento, usabilidade e integração com instituições públicas (Caldeira et al. 2024). Essas lacunas evidenciam a necessidade de sistemas mais inteligentes, acolhedores e sensíveis ao contexto das usuárias, o que reforça a importância do desenvolvimento de um *chatbot* com abordagem centrada na vítima, como o proposto neste trabalho.

Capítulo 4

PROBLEMA

4.1 CONTEXTUALIZAÇÃO DO PROBLEMA

Na Introdução (Capítulo 1), foram apresentados dados que evidenciam a persistência e gravidade da violência contra a mulher no Brasil, incluindo os altos índices de feminicídio e a subnotificação dos casos. Apesar da existência de legislações específicas, como a Lei Maria da Penha, e da crescente adoção de tecnologias de apoio, as vítimas ainda enfrentam barreiras para acessar canais de denúncia de forma rápida, acolhedora e segura.

Esse cenário reforça a necessidade de explorar soluções inovadoras baseadas em tecnologias emergentes, como discutido na Fundamentação Teórica (Capítulo 2), que abordou os avanços em processamento de linguagem natural, chatbots e inteligência artificial aplicada a contextos sociais e sensíveis.

4.2 LIMITAÇÕES DAS SOLUÇÕES EXISTENTES

Conforme descrito em Trabalhos Relacionados (Capítulo 3), já existem iniciativas relevantes, como o PenhaS, o Maria da Penha Virtual e o *chatbot* Aurora. No entanto, a análise desses sistemas revela limitações que comprometem sua eficácia, tais como:

- interfaces pouco intuitivas para usuárias em situação de estresse;
- comunicação rígida e pouco empática, baseada em fluxos fixos de perguntas e respostas;
- ausência de integração completa com órgãos públicos de segurança e acolhimento;
- baixa confiabilidade dos dados coletados devido à interrupção frequente dos relatos.

Essas fragilidades apontam para um descompasso entre as potencialidades das tecnologias de IA e as necessidades reais das mulheres que recorrem a tais ferramentas em momentos críticos.

4.3 FORMULAÇÃO DO PROBLEMA DE PESQUISA

A partir dessa análise, torna-se evidente que as soluções atuais não conseguem equilibrar os aspectos técnicos, sociais e humanos envolvidos no atendimento de ocorrências de violência contra a mulher.

Assim, o problema de pesquisa que orienta este trabalho pode ser formulado da seguinte forma:

Como desenvolver um chatbot inteligente, capaz de realizar atendimento empático, seguro e tecnicamente viável, de modo a acolher e encaminhar ocorrências de violência contra a mulher, superando as limitações identificadas nas soluções atuais?

4.4 HIPÓTESE

Considera-se como hipótese que a integração de modelos de linguagem natural baseados em LLMs, aliados a um fluxo estruturado de coleta de informações e mecanismos de armazenamento seguro, pode oferecer uma experiência de atendimento mais acolhedora. Essa abordagem tem o potencial de aumentar a adesão das usuárias, reduzir interrupções durante o relato e melhorar a qualidade dos dados encaminhados aos órgãos competentes.

Capítulo 5

IMPLEMENTAÇÃO

5.1 VISÃO GERAL DO DESENVOLVIMENTO

O desenvolvimento do **MarIA Um Chatbot Inteligente para Atendimento de Vítimas de Violência contra a Mulher** foi realizado de forma iterativa e flexível, com características de prototipação experimental. Essa abordagem permitiu construir, testar e aprimorar rapidamente componentes do sistema antes de sua consolidação final.

O projeto foi implementado como um conjunto de serviços integrados, orquestrados via *Docker Compose*, incluindo um *bot* para interação com o usuário, agentes LLM para processamento de linguagem natural, um banco de dados relacional e um *dashboard* para visualização de dados e métricas.

Um *dashboard* (ou painel de controle) é uma ferramenta de visualização de dados que exibe, de forma centralizada e gráfica, métricas e Indicadores-Chave de Desempenho. Seu objetivo é permitir o monitoramento e a análise de informações complexas de maneira rápida e intuitiva, facilitando a tomada de decisões (Few 2013). Além disso, foi incorporada inspiração em iniciativas já existentes, como o aplicativo SalveElas, do Rio Grande do Norte, que atua na proteção de mulheres em situação de vulnerabilidade.

5.2 ARQUITETURA E ORGANIZAÇÃO DO PROJETO

O repositório do projeto está disponível publicamente em: <https://github.com/stephannyb/MarIA>

O projeto foi estruturado com base em uma arquitetura de microsserviços, organizada em diferentes módulos encapsulados e orquestrados por meio do *Docker Compose*. Cada serviço é executado em seu próprio contêiner, garantindo isolamento, independência e facilidade de manutenção. Essa escolha possibilita que cada componente possa ser atualizado, substituído ou escalado de forma independente, promovendo maior flexibilidade na evolução do sistema.

A arquitetura é composta por cinco serviços principais: o Bot, responsável pela interface com o usuário via Telegram, recebendo mensagens, estruturando requisições e encaminhando os dados para a API; a API (Agentes), que implementa a lógica de processamento das ocorrências, utilizando agentes especializados em extrair informações

estruturadas; o Banco de Dados (PostgreSQL), que armazena de forma segura as ocorrências processadas; o Dashboard, desenvolvido em Python com Streamlit para visualização de dados; e o PgAdmin, ferramenta de administração do banco de dados, utilizada para consultas e manutenção.

A comunicação entre esses serviços ocorre principalmente via API REST, com o uso de requisições HTTP e troca de dados em formato JSON.

A comunicação entre os serviços é fundamentada em padrões consolidados da web. O estilo arquitetural *REST* (*Representational State Transfer*) define um conjunto de regras para a criação de serviços web interoperáveis. Uma de suas principais características é a comunicação sem estado (*stateless*), onde cada requisição do cliente para o servidor deve conter toda a informação necessária para ser compreendida e processada.

Essas requisições são realizadas utilizando o protocolo *HTTP* (*Hypertext Transfer Protocol*), que define métodos para indicar a ação desejada a ser executada em um recurso. O método *POST*, utilizado neste projeto para o registro de ocorrências, é empregado para submeter dados a um recurso específico, frequentemente resultando na criação de uma nova entidade no servidor. Por fim, os dados trocados entre os serviços são estruturados no formato *JSON* (*JavaScript Object Notation*), um padrão textual leve e de fácil leitura tanto para humanos quanto para máquinas, ideal para a serialização e transmissão de dados em APIs. (Fielding 2000)

Essa padronização garante interoperabilidade e facilita a integração futura com outros sistemas.

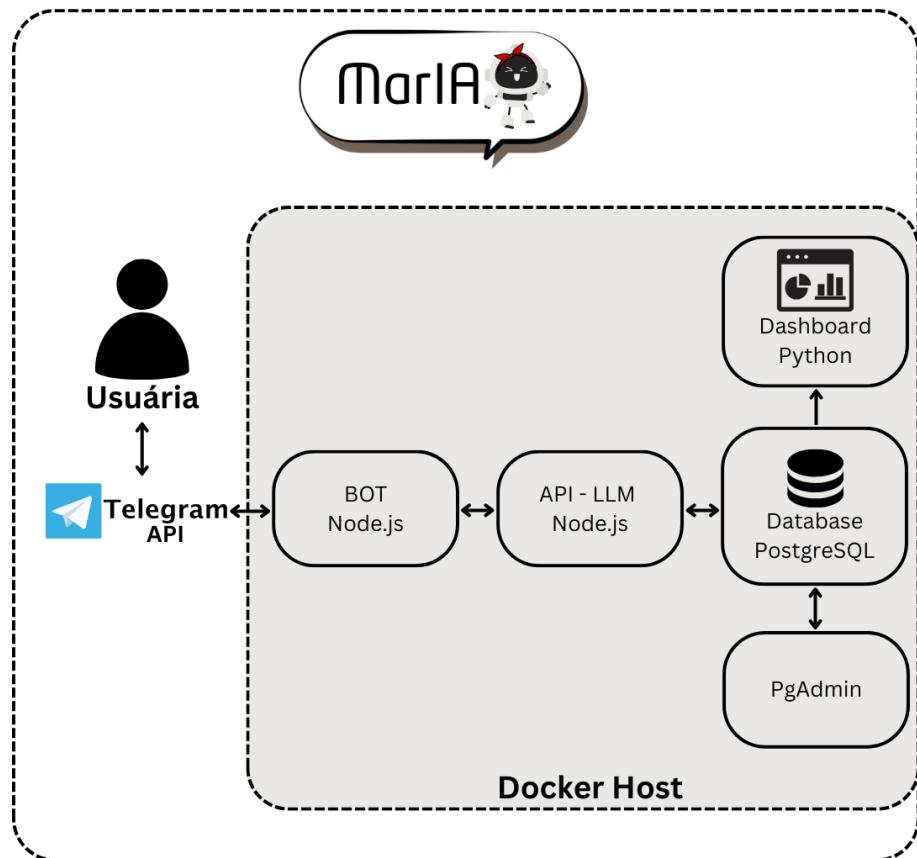


Figura 5.1: Arquitetura do sistema MarIA.

Fonte: Produção da Autora.

5.2.1 Comunicação via API REST

A API expõe dois *endpoints* principais:

- /api/maria/ocorrencia/registrar (método POST): responsável por receber um relato inicial da ocorrência enviado pelo *bot* no *Telegram*. Os dados recebidos incluem informações como texto do relato, possíveis anexos (fotos, localização), além de sinalizações contextuais (como presença de agressor armado ou ocorrência em andamento). O *endpoint* retorna um JSON estruturado contendo os dados extraídos pelo módulo de agentes.
- /api/maria/ocorrencia/salvar (método POST): utilizado como redundância no processo de registro. Caso haja falha no armazenamento direto da ocorrência,

esse endpoint recebe o JSON previamente gerado e realiza a tentativa de persistência no banco de dados.

O formato da requisição segue o padrão `application/json`, com parâmetros específicos, como exemplificado abaixo:

```
{
  "nomeUsuario": "Usuário do Telegram",
  "textoInformal": "Exemplo de relato...",
  "fotos": [],
  "localizacoes": [],
  "agressorArmado": false,
  "acontecendoAgora": true
}
```

A resposta segue um modelo padronizado com os campos `success`, `message` e `data`, permitindo identificar facilmente o status da requisição:

```
{
  "success": true,
  "message": "Ocorrência registrada com sucesso.",
  "data": { ... }
}
```

O objeto `data` contém os dados estruturados da ocorrência, incluindo informações de localização, data e hora, dados da vítima, envolvidos, testemunhas e tipificação do crime (violência física, psicológica, sexual, patrimonial, entre outros).

5.2.2 Justificativa do Uso de Microsserviços com *Docker*

A adoção da arquitetura de microsserviços orquestrada com Docker foi motivada por uma série de vantagens técnicas e operacionais. Entre elas estão o isolamento, pois cada módulo é executado em um contêiner independente, evitando que falhas impactem os demais. Destaca-se também a escalabilidade, permitindo que serviços críticos, como a API, sejam replicados horizontalmente, e a facilidade de manutenção, já que atualizações em um serviço não exigem a reinicialização de toda a aplicação. Adicionalmente, a abordagem garante portabilidade, permitindo a execução consistente em diferentes ambientes, e promove uma integração simplificada pelo uso do Docker Compose para orquestrar os serviços.

Essa abordagem garante que o projeto esteja preparado para evoluir, seja pela adição de novos serviços (como módulos de análise de dados ou integração com serviços governamentais), seja pelo aumento de demanda de usuários.

5.2.3 Estrutura dos Dados da API

Além da definição dos *endpoints*, foi projetada uma estrutura padronizada para troca de informações entre os módulos, utilizando o formato JSON como meio de serialização. Essa padronização garante consistência nas mensagens, facilita o tratamento de erros e possibilita futuras integrações com sistemas externos.

Resposta HTTP Padrão

Uma Resposta HTTP Padrão, no contexto de uma API REST, é um modelo de estrutura de dados consistente retornado pelo servidor após o processamento de uma requisição. Adotar um padrão é uma boa prática no desenvolvimento de software, pois torna a comunicação entre o cliente (neste caso, o *bot*) e o servidor mais previsível e robusta. A estrutura geralmente inclui um indicador de sucesso da operação (*booleano*), uma mensagem descritiva para o usuário ou desenvolvedor e, em caso de sucesso, o corpo dos dados solicitados. Essa abordagem simplifica o tratamento de sucessos e erros de forma uniforme em toda a aplicação (Richardson & Ruby 2007).

```
{  
    "success": true,  
    "message": "Ocorrência registrada com sucesso.",  
    "data": { ... }  
}
```

- **success**: indica o status da requisição (*booleano*).
- **message**: mensagem de *feedback* sobre a operação.
- **data**: objeto contendo os dados estruturados da ocorrência, ou *null* em caso de falha.

Estrutura do JSON da Ocorrência

O objeto **data** segue o padrão abaixo:

- **dados_localizacao**: informações sobre o local da ocorrência (endereço, município, coordenadas geográficas).
- **dados_data_hora**: datas e horários do registro e do acontecimento.
- **dados_vitima**: informações da vítima (nome, idade, contatos).
- **dados_testemunhas**: lista de testemunhas com dados básicos de identificação e contato.
- **dados_envolvidos**: lista de pessoas envolvidas, incluindo condição (vítima, agressor, etc.).
- **dados_crime**: tipificação da violência, incluindo subdivisões (física, moral, psicológica, sexual, patrimonial e feminicídio).
- **texto_narrativa**: relato textual da ocorrência.
- **observacao**: informações adicionais fornecidas.

Exemplo de Estrutura

A seguir, um exemplo da organização dos campos:

```
{
  "dados_localizacao": {
    "tipo_local": "residência",
    "nome_municipio": "Natal",
    "sigla_uf": "RN",
    "endereco_logradouro": "Rua Exemplo",
    "endereco_numero": "123",
    "endereco_bairro": "Centro",
    "longitude": "-35.2075",
    "latitude": "-5.7945"
  },
  "dados_data_hora": {
    "data_registro_ocorrencia": "2025-09-01",
    "horario_registro_ocorrencia": "14:32",
    "data_acidente_ocorrencia": "2025-09-01",
    "horario_acidente_ocorrencia": "13:50"
  },
  "dados_vitima": {
    "nome": "Maria Silva",
    "idade": "28",
    "tel_contato": "(84) 99999-0000"
  },
  "dados_envolvidos": [
    {
      "nome": "João Souza",
      "condicao_envolvido": "suposto agressor"
    }
  ],
  "dados_crime": {
    "violencia_fisica": [
      {
        "agressao_fisica": true,
        "dados_data": "2025-09-01",
        "dados_hora": "13:50"
      }
    ],
    "violencia_psicologica": [
      {
        "ameacas": true,
        "humilhação": false
      }
    ]
  }
}
```

```

        ],
        "feminicidio": null
    },
    "texto_narrativa": "Relato textual enviado pela vítima.",
    "observacao": null
}

```

Estrutura Detalhada dos Objetos

A seguir são apresentados os principais objetos que compõem o JSON da ocorrência:

- **dados_localizacao**: inclui parâmetros como `tipo_local`, `nome_municipio`, `sigla_uf`, `longitude` e `latitude`.
- **dados_data_hora**: registra data e hora do registro e do acontecimento.
- **dados_vitima**, **dados_testemunhas** e **dados_envolvidos**: armazenam informações pessoais (nome, idade, documento de identificação, contatos).
- **dados_crime**: subdividido em `violencia_fisica`, `violencia_moral`, `violencia_psicologica`, `violencia_sexual`, `violencia_patrimonial` e `feminicidio`, cada qual com indicadores booleanos e registros de data/hora.

Essa modelagem garante que os relatos informais enviados pelas vítimas ao *chatbot* sejam transformados em registros estruturados, com potencial para posterior análise estatística, integração com bancos de dados policiais ou utilização em estudos sobre violência de gênero.

5.3 SERVIÇOS E INFRAESTRUTURA

A aplicação é composta pelos seguintes serviços definidos no `docker-compose.yml`: *Bot*, implementado em *JavaScript* para gerenciar a interface de conversação; Agentes, também em *JavaScript*, que processam requisições assíncronas de linguagem natural; *PostgreSQL* (`db`), utilizado como banco de dados relacional para armazenamento; *PgAdmin*, como ferramenta para administração do banco de dados; e o *Dashboard*, desenvolvido em *Streamlit (Python)* para análise e visualização dos dados coletados.

5.4 FERRAMENTAS E TECNOLOGIAS

O desenvolvimento utilizou as seguintes tecnologias: *JavaScript*, usado para o *bot* e para os agentes *LLM*, pela eficiência no tratamento de requisições assíncronas; *Python* em conjunto com *Streamlit*, utilizados no *dashboard* pela simplicidade de prototipação e integração com bibliotecas de visualização; *PostgreSQL*, como banco de dados relacional para armazenamento estruturado; e *Docker Compose*, para a orquestração de múltiplos serviços de forma integrada.

5.5 IMPLEMENTAÇÃO E MÓDULOS

A implementação do sistema foi estruturada em módulos independentes, de forma a favorecer a organização do código, a escalabilidade e a clareza na manutenção. Cada módulo desempenha um papel específico na arquitetura geral do *chatbot*, contribuindo para o funcionamento integrado da solução. A seguir, são apresentados os principais módulos desenvolvidos.

5.5.1 Módulo de PLN (API)

O módulo de Processamento de Linguagem Natural constitui o núcleo do sistema, sendo responsável pela interpretação e transformação dos relatos recebidos. Ele foi desenvolvido utilizando a biblioteca *LangChain* em conjunto com o modelo GPT, fornecido pela API da organização *OpenAI*.

A estrutura desse módulo parte de um arquivo base denominado `baseLLM.js`, no qual é configurada a instância central do modelo de linguagem. Essa instância serve de fundação para os demais agentes especializados. O código é o seguinte:

Modelo Base

O arquivo `baseLLM.js` define a configuração comum a todos os agentes, incluindo o modelo a ser utilizado, a chave de autenticação e parâmetros como a *temperature*, que foi fixada em zero para garantir respostas determinísticas e consistentes:

```
import { ChatOpenAI } from "@langchain/openai";

export const chatGPTModel = new ChatOpenAI({
    model: process.env.OPENAI_MODEL,
    apiKey: process.env.OPENAI_API_KEY,
    temperature: 0,
});
```

Agente de Extração Genérica de Dados

O agente genérico (`genericDataAgent`) tem como objetivo extrair informações estruturadas a partir do relato fornecido pela vítima, incluindo dados da vítima, do suposto agressor, testemunhas, local, data, hora e crimes cometidos.

O esquema de saída foi definido com *Zod*, uma biblioteca de declaração e validação de esquemas para *TypeScript* e *JavaScript*. Sua principal finalidade é garantir a integridade e a segurança de tipos de dados em tempo de execução. Com o *Zod*, é possível definir uma estrutura rigorosa (um *schema*) para um objeto de dados, como um JSON, e validar se os dados recebidos estão em conformidade com essa estrutura. Caso contrário, a biblioteca impede que dados malformados ou inesperados sejam processados pela aplicação, tornando o sistema mais previsível e menos suscetível a erros (Colin 2024). No contexto

deste trabalho, o Zod foi utilizado para assegurar que a resposta da API do LLM seguisse estritamente o formato JSON esperado.

Um exemplo representativo do *prompt* utilizado está associado à detecção de crimes:

Extraia os tipos de violência mencionados na ocorrência.

Retorne apenas os nomes das violências listadas abaixo,
sem adicionar parênteses ou palavras extras:

- Violencia Fisica
- Violencia Moral
- Violencia Psicologica
- Violencia Sexual
- Violencia Patrimonial
- Feminicidio

Exemplo:

Entrada: "Meu companheiro me empurrou e rasgou meus documentos."

Saída esperada: ["Violencia Fisica", "Violencia Patrimonial"]

Esse agente é responsável por fornecer a base de dados estruturados que alimenta os demais módulos do sistema.

Agente Escrivão (*ScrivenerAgent*)

O segundo agente desenvolvido, denominado *scrivenerAgent*, atua como um escrivão da Polícia Civil. Seu objetivo é reescrever o relato em norma culta da língua portuguesa, corrigindo erros gramaticais, ortográficos e regionais, sem alterar o sentido do texto. Isso assegura que a narrativa registrada seja compatível com documentos oficiais.

Você é um escrivão profissional da Polícia Civil. Você tem o papel de ler textos e corrigi-los para a norma culta da língua portuguesa, sem fazer alterações no sentido da história.

Agente Especializado em Violência Contra a Mulher

O terceiro agente, denominado *specializedAgent*, é especializado na análise de ocorrências de violência contra a mulher. Ele classifica os diferentes tipos de violência previstos na legislação brasileira (física, moral, psicológica, sexual, patrimonial e feminicídio), preenchendo um esquema estruturado de saída.

O *prompt* que orienta o agente é o seguinte:

Você é um perito analista de textos profissional da Polícia Civil.
Você é especializado na extração de dados de ocorrências de violência contra a mulher. Sua função é identificar quais os tipos de violência contra a mulher relatados e verificar se houve ou não um feminicídio.
Você deve relatar os tipos de violência ocorridos e caso não haja dados referentes a algum deles, retorne null.

Esse módulo representa o ponto central de especialização do sistema, pois permite que o *chatbot* identifique automaticamente as dimensões da violência reportada, gerando um retrato fiel e estruturado da ocorrência.

Integração dos Agentes

O fluxo de processamento segue três etapas principais:

1. O `scrivenerAgent` corrige o relato da vítima para a norma culta;
2. O `genericDataAgent` extrai dados estruturados gerais (vítima, suposto agressor, testemunhas, local, data, hora e crimes);
3. O `specializedAgent` analisa especificamente os tipos de violência contra a mulher.

A combinação desses agentes garante que o sistema seja capaz de transformar um relato em linguagem natural em uma representação estruturada e formalizada, apta para integrar sistemas de apoio às forças de segurança e serviços de acolhimento.

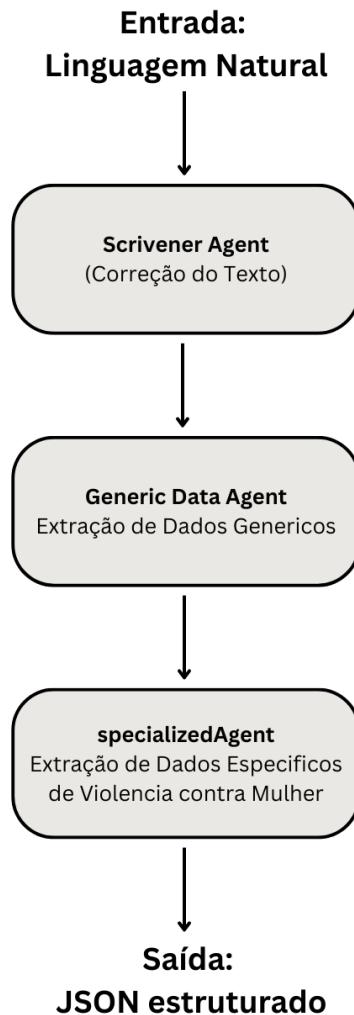


Figura 5.2: Fluxograma de processamento de um relato pelos agentes de PLN.
Fonte: Produção da Autora.

5.5.2 Módulo de Integração com o *Telegram* (*BOT*)

O módulo de integração com o *Telegram* constitui a camada de interface entre o usuário e o sistema. Ele foi desenvolvido utilizando a biblioteca `node-telegram-bot-api`, que permite a criação de robôs de conversa (*chatbots*) de forma simples e escalável. O *bot* opera em modo de *polling*, recebendo e processando mensagens em tempo real.

O arquivo principal do *bot* é responsável por instanciar o objeto `TelegramBot`, configurar a autenticação com o *token* fornecido pela plataforma e inicializar as rotinas de registro de usuários e processamento de mensagens. Um trecho representativo da configuração inicial é o seguinte:

```
const TelegramBot = require("node-telegram-bot-api");
```

```
const token = process.env.BOT_TOKEN;
const bot = new TelegramBot(token, { polling: true });
```

Funcionalidades de Entrada

O *bot* foi projetado para aceitar múltiplas modalidades de entrada, de forma a garantir acessibilidade e flexibilidade ao registro de ocorrências. Entre os recursos disponíveis estão:

- **Texto:** entrada textual tradicional, analisada diretamente pelos agentes de PLN.
- **Áudio:** mensagens de voz enviadas pelo usuário são convertidas de `.ogg` para `.wav` utilizando `ffmpeg`, e em seguida transcritas por meio da API do *Google Cloud Speech-to-Text*.
- **Imagens:** arquivos de imagem podem ser submetidos pelo usuário. Nestes casos, aplica-se Reconhecimento Óptico de Caracter (*Optical Character Recognition OCR*) com a biblioteca `node-tesseract-ocr`, permitindo recuperar trechos textuais embutidos na mídia.
- **Localização:** a API do *Telegram* possibilita o envio da localização geográfica diretamente pelo aplicativo, que é armazenada e vinculada ao registro da ocorrência.

Fluxo de Conversa e Interação

A interação com a vítima é conduzida por meio de mensagens automáticas e botões de resposta rápida (`inline_keyboard`). O fluxo inicial de conversa apresenta saudações e opções de ação:

```
const askWhatToDo = {
    text: "O que você deseja fazer agora?",
    options: {
        reply_markup: {
            inline_keyboard: [
                [
                    { text: "Registrar uma ocorrência", callback_data: "register" },
                    { text: "Quero saber mais", callback_data: "more_info" }
                ]
            ]
        }
    }
};
```

Esse recurso simplifica a tomada de decisão do usuário e garante que as informações sejam coletadas de forma organizada.

Integração com os Agentes

Ao final do processo de coleta de informações, o *bot* envia os dados para a API que centraliza os agentes de PLN. Esse envio é feito via requisição HTTP com a biblioteca `axios`:

```
async function callAgents(payload) {
    return await axios.post(
        "http://agentes:3000/api/maria/ocorrecia/registrar",
        payload
    );
}
```

Com isso, garante-se que os relatos coletados sejam processados, normalizados e classificados automaticamente, possibilitando o registro estruturado das ocorrências.

Recursos de Persistência

Para manter o estado da interação com os usuários, o sistema utiliza um banco de dados *PostgreSQL*. Dois esquemas principais foram definidos:

- **greeted_users**: armazena o estado de cada usuário em relação ao fluxo de interação (edição em andamento, aguardando mensagem, etc.);
- **messages**: armazena os relatos e mídias coletados durante as interações.

Essa camada de persistência garante que a experiência da vítima seja contínua, mesmo em casos de interrupção do fluxo ou reinício do *bot*.

Diferenciais de Projeto

Alguns recursos foram incorporados ao *bot* com o objetivo de aumentar a utilidade prática do sistema:

- **Palavras-chave**: termos como “pizza” ou “açaí” são interpretados como indicativos de que a violência está ocorrendo no momento do contato, disparando fluxos de resposta imediata.
- **Confirmação progressiva**: o *bot* apresenta ao usuário um resumo parcial das informações já coletadas e pergunta se deseja complementar, evitando perda de dados importantes.
- **Supporte multimodal**: integração nativa de texto, áudio, imagens e localização em um único fluxo conversacional.

Esse módulo representa a face visível do sistema, permitindo que a vítima interaja de maneira simples, segura e multimodal com o *chatbot* por meio do *Telegram*.

5.5.3 Módulo de Banco de Dados

O banco de dados adotado no sistema é o *PostgreSQL*, escolhido pela robustez, compatibilidade com bibliotecas modernas e suporte a operações complexas. Ele é responsável por armazenar tanto os dados estruturados das ocorrências quanto o estado de interação com os usuários do *chatbot*.

A arquitetura de dados, ilustrada na Figura 5.3, foi projetada para garantir a persistência e a recuperação do fluxo conversacional. Ela é composta por três tabelas principais. A tabela `greeted_users` que armazena o `chat_id` de cada usuário e sinalizadores de estado, `in_edition` e `waiting_message`. Sua função é permitir que, em caso de interrupção ou reinício do *bot*, o sistema possa identificar se uma conversa estava em andamento. Para dar continuidade, o *bot* consulta a tabela `messages`, que armazena o conteúdo bruto, texto, áudio, localização, de todas as interações trocadas.

Finalmente, a tabela `finised_ocs` armazena a ocorrência consolidada e estruturada, somente após ela ter sido processada pelos agentes de PLN. É esta tabela que serve como fonte de dados primária para as análises e visualizações apresentadas no *dashboard*. Além disso, a utilização do *pgAdmin*, acessível via contêiner Docker, facilita o gerenciamento, inspeção e validação dos dados durante o desenvolvimento.

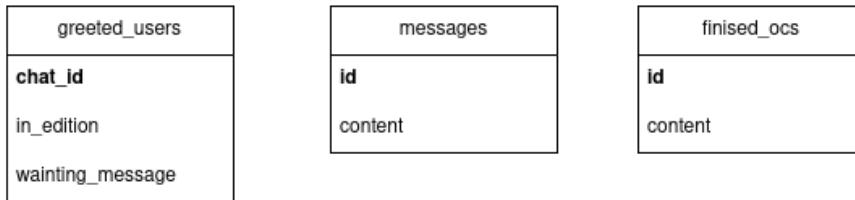


Figura 5.3: Diagrama de entidade e relacionamento do banco de dados.

Fonte: Produção da Autora.

5.5.4 Módulo de Dashboard

O *dashboard* de monitoramento foi implementado em Python utilizando a biblioteca Streamlit, permitindo a criação rápida de uma interface interativa para a análise dos dados provenientes da tabela `finised_ocs`. Esse módulo tem caráter complementar, servindo como apoio para pesquisadores e profissionais que possam vir a acompanhar o sistema.

Conforme detalhado no Capítulo 6, a interface permite a filtragem de ocorrências por período de registro, tipo de crime, nome da vítima ou palavras-chave presentes no relato. A interface foi desenhada para fornecer três níveis de análise:

Primeiro, uma análise geoespacial, que exibe um mapa interativo com a distribuição geográfica das ocorrências, permitindo a identificação de pontos de calor de violência.

Segundo, uma análise estatística e temporal, que apresenta o número total de registros, gráficos de linha sobre a evolução das ocorrências ao longo dos meses, e gráficos de barras que classificam os tipos de crime mais frequentes.

Terceiro, uma análise de padrões e relações, que inclui a distribuição de ocorrências por dia da semana e um *heatmap* (mapa de calor) de coocorrência, que ilustra a frequência com que diferentes tipos de violência são relatados conjuntamente. Por meio dele, as informações coletadas são apresentadas de forma clara, favorecendo a análise situacional e o suporte à tomada de decisão.

5.6 INTEGRAÇÃO E TESTES

A integração consistiu na conexão dos módulos de PLN e do *Bot* no *Telegram*, garantindo que as informações fluíssem corretamente entre eles. O processo de testes foi realizado de forma incremental, iniciando com testes unitários para validar cada módulo de forma isolada. Em seguida, foram executados testes integrados para verificar a comunicação entre os diferentes módulos do sistema.

Por fim, a etapa de testes funcionais simulou interações reais com usuários, o que permitiu avaliar a clareza, o tempo de resposta e a adequação das mensagens do *chatbot*. Todos os testes foram realizados em ambiente controlado, utilizando contas de teste no *Telegram* e cenários simulados de conversas. Essa abordagem permitiu identificar e realizar os ajustes necessários antes de uma eventual implantação futura.

Capítulo 6

RESULTADOS

6.1 RESULTADOS FUNCIONAIS

Nesta seção são apresentados os resultados funcionais do chatbot, demonstrando a validação prática dos objetivos específicos delineados. Os testes validaram que o sistema é capaz de interagir de forma multimodal no Telegram, oferecendo acolhimento e conduzindo o registro da ocorrência de forma eficaz. Adicionalmente, foi comprovado o funcionamento de fluxos diferenciais, projetados para fortalecer o vínculo com a usuária.

6.1.1 Fluxo de Interação com a Usuária

O contato inicial da usuária com o chatbot se dá por meio de mensagens de acolhimento e da apresentação das opções de ajuda. As Figuras 6.1 e 6.2 demonstram este fluxo, no qual o sistema oferece orientações e informações complementares.

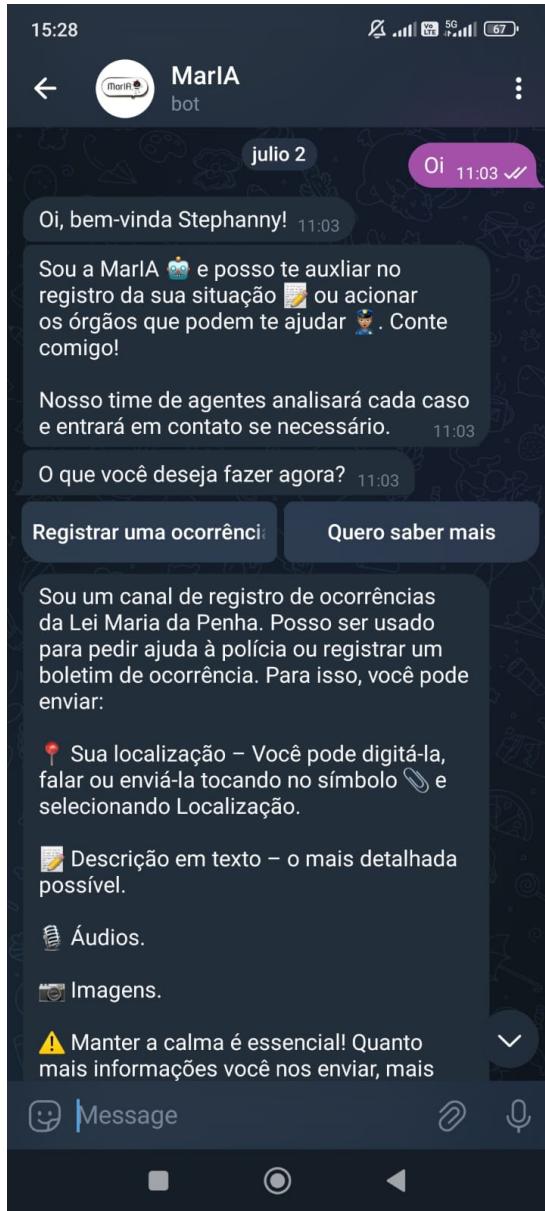


Figura 6.1: Mensagem inicial de acolhimento.



Figura 6.2: Fluxo informativo.

6.1.2 Registro Multimodal da Ocorrência

Um dos objetivos específicos do sistema foi possibilitar diferentes formas de relato, garantindo acessibilidade e flexibilidade. Os resultados mostraram que esse objetivo foi plenamente atendido.

A usuária pode relatar um evento tanto por meio de áudio, que é automaticamente transcrito pelo sistema (Figura 6.3), quanto por texto livre (Figura 6.4). Adicionalmente, é possível anexar a localização geográfica (Figura 6.5) e imagens (Figura 6.6) para enriquecer o registro.

6.1. RESULTADOS FUNCIONAIS

35



Figura 6.3: Registro via áudio com transcrição.

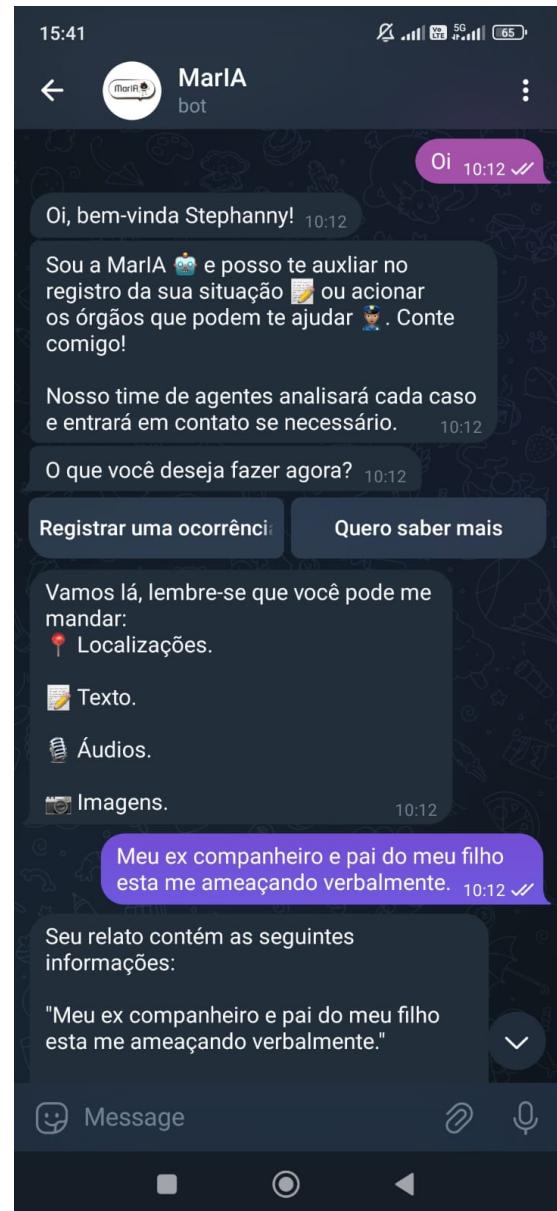


Figura 6.4: Registro via mensagem de texto.



Figura 6.5: Compartilhamento de localização.

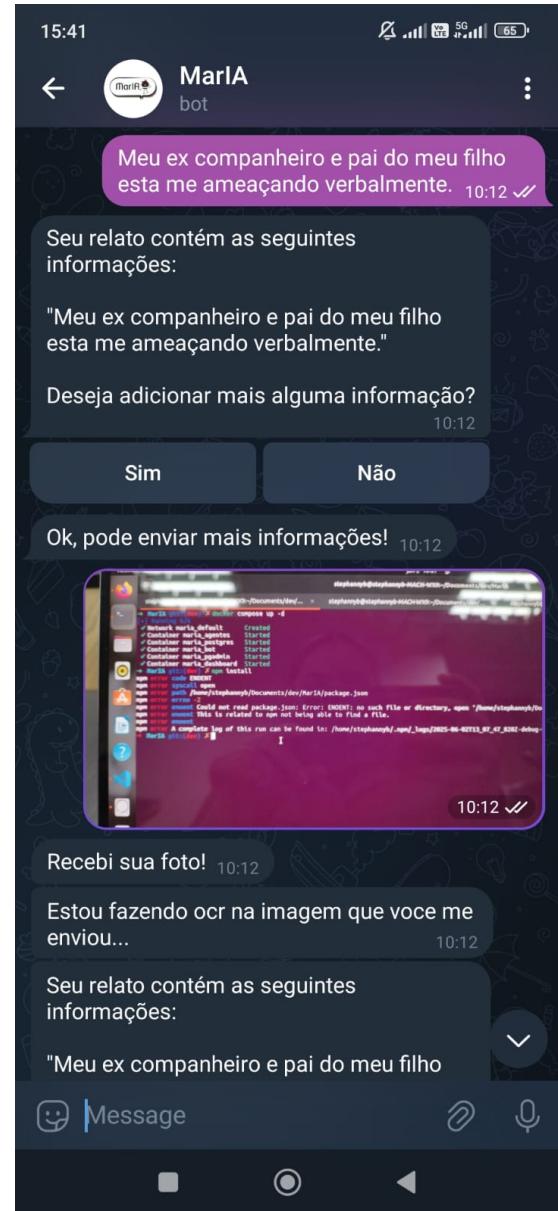


Figura 6.6: Envio de imagem anexa.

Ao final, o sistema consolida as informações coletadas e confirma a conclusão do registro, como ilustrado na Figura 6.7.

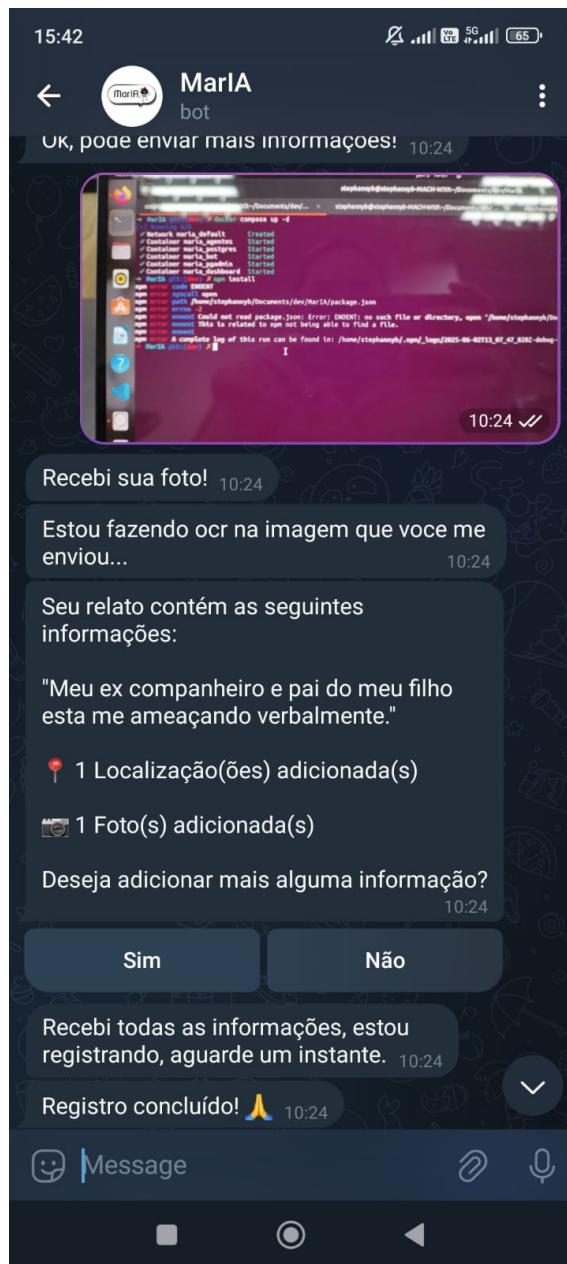
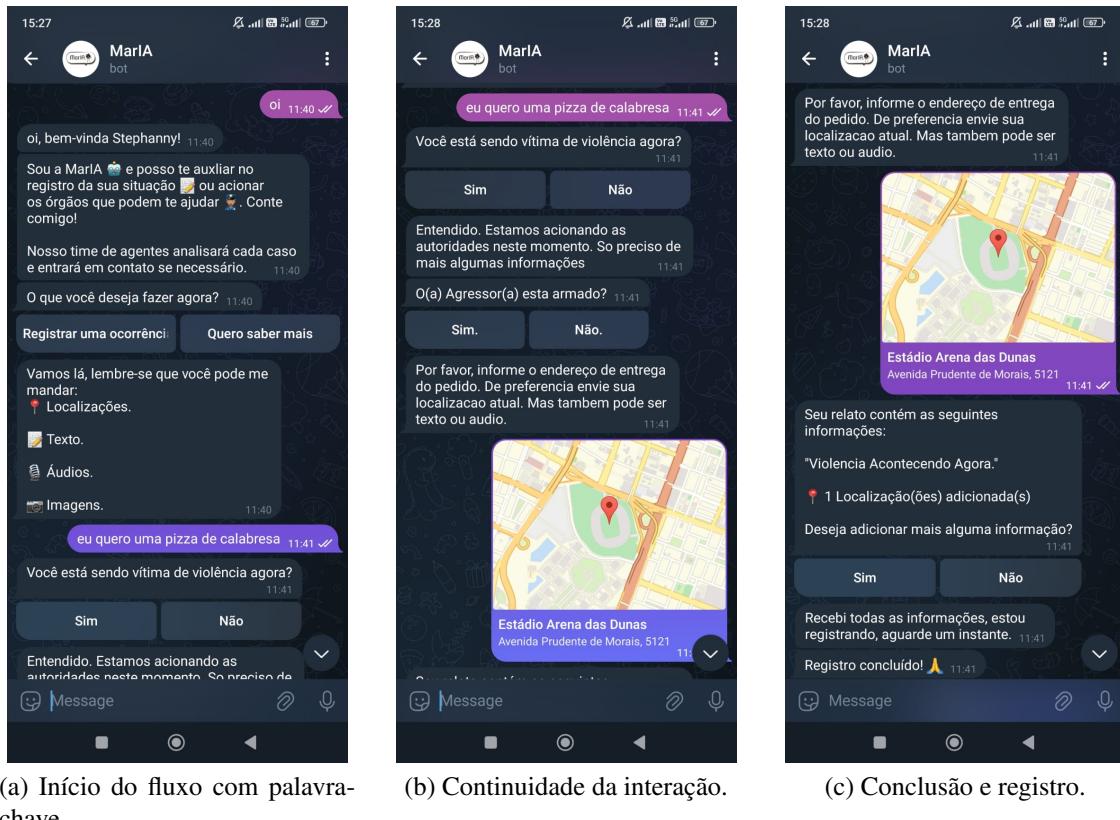


Figura 6.7: Etapa de conclusão do registro da ocorrência.

6.1.3 Fluxos Diferenciais

Outro objetivo específico foi oferecer alternativas de interação usando palavras-chave que, embora não estejam diretamente ligadas ao registro de violência, contribuem para manter em situações de emergência de forma discreta. Esses fluxos diferenciais são ativados quando a usuária utiliza determinadas palavras-chave, como “pizza” ou “açaí”, simulando um pedido o que ajuda a manter a discrição e o sigilo.

As Figuras 6.8a, 6.8b e 6.8c mostram o exemplo de um desses fluxos em operação.



(a) Início do fluxo com palavra-chave.

(b) Continuidade da interação.

(c) Conclusão e registro.

Figura 6.8: Exemplo de um fluxo de interação diferencial, ativado por palavra-chave, mostrando o início, a continuidade e a conclusão do registro de emergência.

6.1.4 Síntese dos Resultados

Com base nos testes realizados, observa-se que todos os objetivos específicos foram atendidos. O *chatbot* demonstrou capacidade de acolher e orientar a usuária no início da interação. Além disso, o sistema permitiu múltiplas modalidades de registro da ocorrência (texto, áudio, imagem e localização) e conseguiu estruturar e concluir o registro através da API dos agentes *LLM*, confirmando a operação. Por fim, foram validados os fluxos diferenciais, projetados para situações de emergência que exigem discrição.

Dessa forma, o sistema demonstrou-se funcional e aderente às metas estabelecidas, comprovando a viabilidade da proposta.

6.2 DASHBOARDS E MONITORAMENTO

Além da interação direta pelo *chatbot*, um dos resultados funcionais do projeto é o *dashboard* de monitoramento, que permite o acompanhamento estruturado das informações coletadas. Essas visualizações fornecem suporte à tomada de decisão e facilitam a análise estatística das ocorrências registradas.

A interface principal, apresentada na Figura 6.9, foca na Localização e Detalhes das Ocorrências. Ela apresenta um mapa interativo que exibe a distribuição geográfica dos relatos e uma tabela com as ocorrências mais recentes. No painel lateral, é possível aplicar filtros dinâmicos por data, tipo de crime, nome da vítima ou palavras-chave.

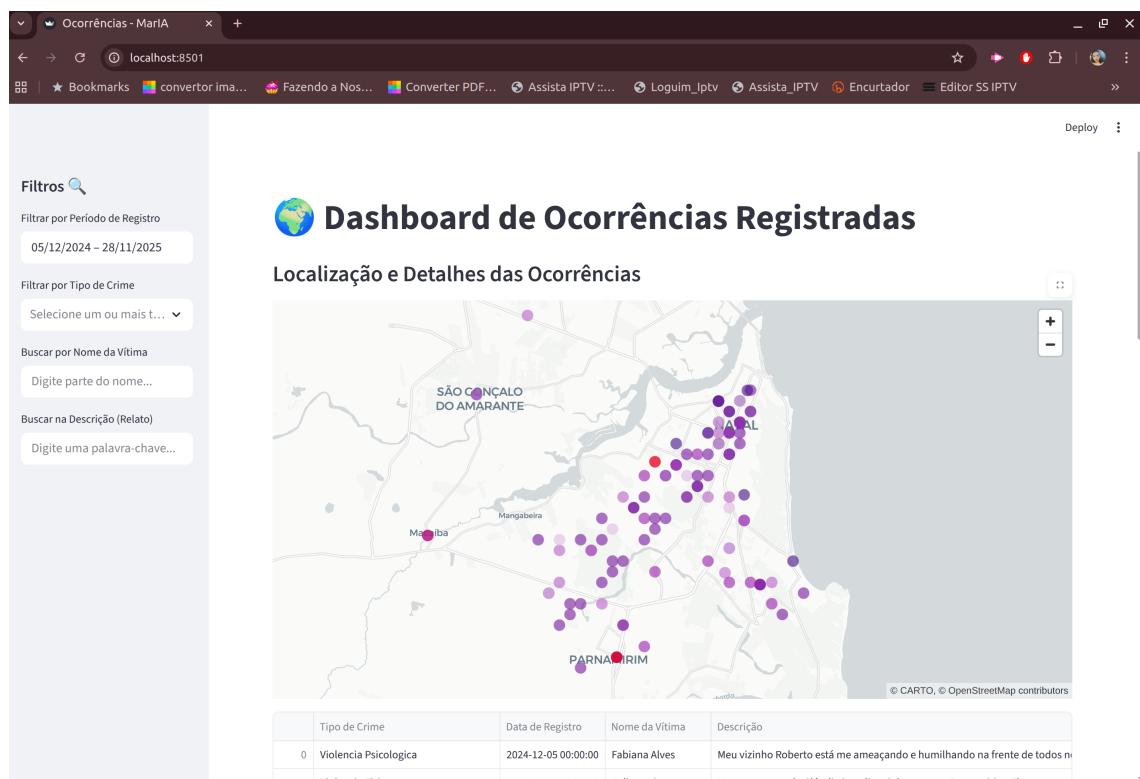


Figura 6.9: Tela principal do Dashboard Análise geoespacial e listagem de ocorrências.
Fonte: Produção da Autora.

Ao rolar a página, a usuária acessa a Análise dos Dados Filtrados (Figura 6.10). Esta seção apresenta métricas agregadas, como o total de ocorrências, um gráfico de linha que demonstra a variação de registros ao longo dos meses, e um gráfico de barras com os tipos de crime mais frequentes.

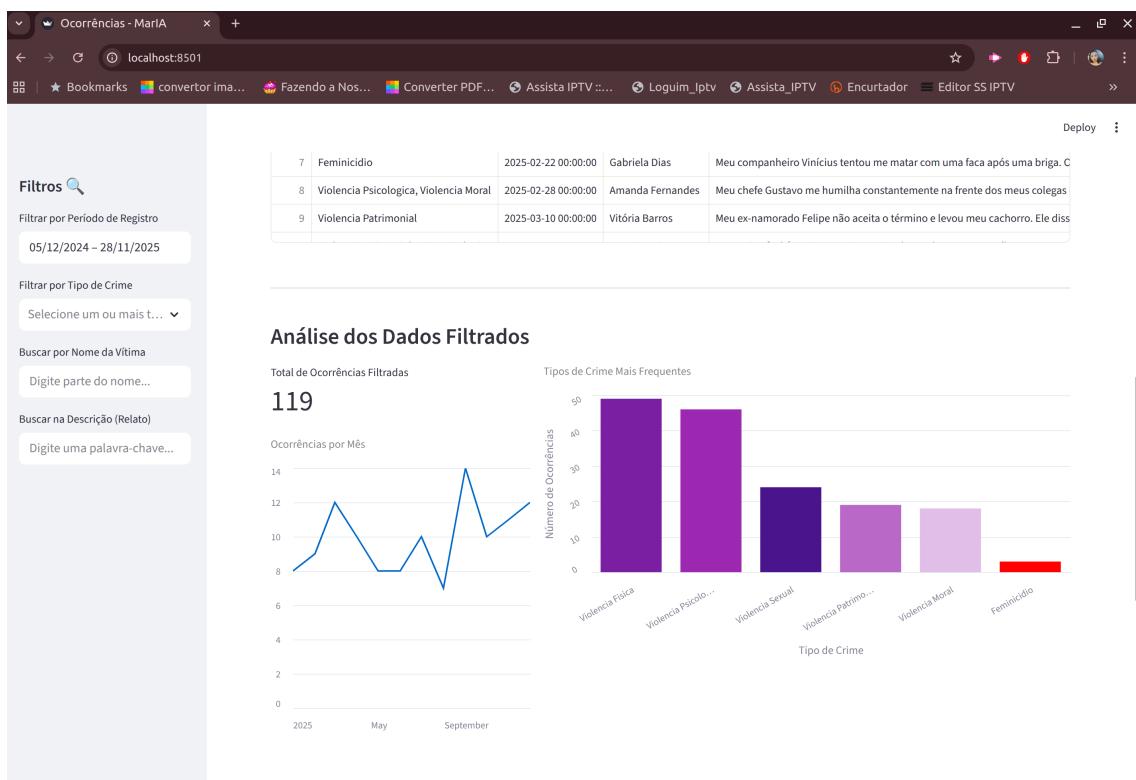


Figura 6.10: Dashboard Análise estatística e temporal.

Fonte: Produção da Autora.

Finalmente, a Figura 6.11 exibe a seção de Padrões Temporais e Relações entre Tipos de Violência. O gráfico de barras à esquerda revela a distribuição das ocorrências por dia da semana. À direita, um *heatmap* de coocorrência espelhada demonstra a relação entre os tipos de violência, mostrando, por exemplo, que diferentes tipos de violência são relatados conjuntamente.

6.3. LIMITAÇÕES E DIFICULDADES

41

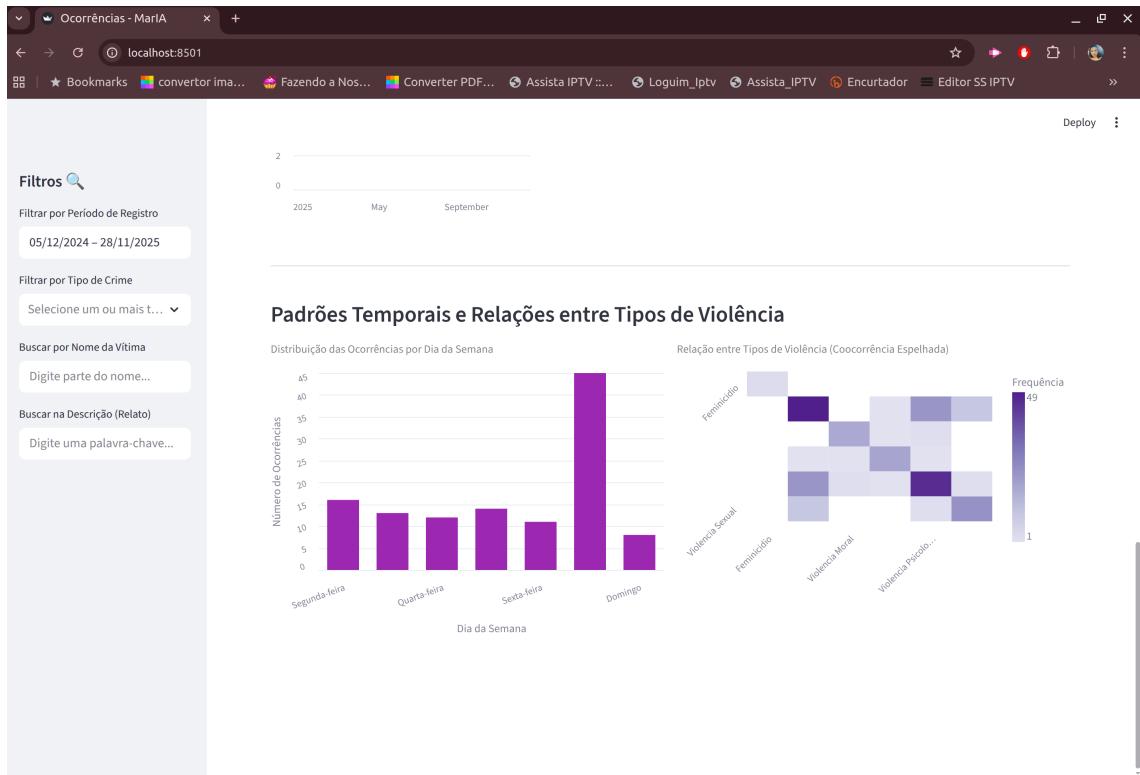


Figura 6.11: Dashboard Análise de padrões por dia da semana e coocorrência de crimes.
Fonte: Produção da Autora.

6.3 LIMITAÇÕES E DIFICULDADES

Durante a validação prática, algumas dificuldades foram observadas. A transcrição de áudio em ambientes com ruído, utilizando a versão de teste da *API Google Speech-to-Text*, apresentou limitações na acurácia da transcrição quando os áudios continham interferências externas. De forma similar, o reconhecimento de texto em imagens de baixa qualidade mostrou dificuldades, com a aplicação de *OCR (Optical Character Recognition)* falhando em identificar corretamente informações em fotos desfocadas ou com pouca iluminação. Finalmente, o agente de extração de dados conseguiu identificar corretamente os tipos de violência em X

Apesar dessas limitações, os resultados demonstram que o sistema é funcional e cumpre o papel de apoiar o registro inicial das ocorrências, além de oferecer uma interação acessível e multimodal.

6.4 VALIDAÇÃO DOS OBJETIVOS ESPECÍFICOS

A validação dos objetivos específicos foi realizada por meio da verificação prática das funcionalidades implementadas. O primeiro objetivo, referente ao desenvolvimento do *chatbot*, foi atingido com a implementação operacional do sistema no *Telegram*, per-

mitindo registros multimodais (texto, áudio, imagem e localização) e assegurando uma comunicação direta e privada.

O segundo objetivo foi cumprido com a integração bem-sucedida da API de PLN baseada em LLM, que demonstrou capacidade para interpretar os relatos e classificar as ocorrências conforme os tipos de violência previstos.

Por fim, o terceiro objetivo foi validado pela estruturação dos dados em um banco relacional e sua posterior apresentação em *dashboards* analíticos, que possibilitam a visualização de métricas e oferecem suporte à tomada de decisão por parte das instituições competentes.

Capítulo 7

CONCLUSÃO

Este trabalho buscou responder ao problema de como desenvolver uma ferramenta tecnológica capaz de oferecer atendimento inicial empático, seguro e eficiente a vítimas de violência contra a mulher, superando limitações de interfaces rígidas e pouco acolhedoras. A proposta central foi o desenvolvimento do MarIA, um *chatbot* inteligente integrado ao *Telegram*, que utiliza Modelos de Linguagem de Grande Escala (LLM) para interpretar relatos e estruturar ocorrências.

7.1 RESPOSTA AO PROBLEMA DE PESQUISA

Os resultados obtidos em ambiente controlado demonstram a viabilidade técnica e funcional da solução proposta. O sistema foi capaz de realizar atendimentos simulados de forma humanizada, interpretando linguagem natural e aceitando múltiplas formas de entrada (texto, áudio, imagem), o que reduz barreiras de comunicação em momentos de estresse. A integração com LLMs permitiu a classificação automática dos tipos de violência, agilizando o processo de triagem sem perder a sensibilidade necessária ao tema.

Portanto, a hipótese inicial de que a união entre interfaces conversacionais acessíveis e inteligência artificial generativa pode melhorar a experiência de denúncia foi corroborada pelos testes funcionais. A solução mostrou-se tecnicamente viável e com potencial para preencher as lacunas identificadas nas ferramentas atualmente disponíveis.

7.2 CONTRIBUIÇÕES

As principais contribuições deste estudo incluem a validação de uma arquitetura de microsserviços para sistemas sensíveis, garantindo modularidade e segurança no tratamento de dados. Além disso, o trabalho entregou um protótipo funcional que não apenas coleta relatos, mas os estrutura de forma útil para a formulação de políticas públicas, como demonstrado pelos painéis de monitoramento desenvolvidos. O projeto também contribui com a discussão sobre o uso ético de IA em contextos sociais críticos.

7.3 LIMITAÇÕES

É importante ressaltar que a validação ocorreu em cenários simulados, não tendo sido o sistema submetido ao imprevisível ambiente de produção real com vítimas reais. Limitações técnicas também foram observadas, como a dependência da qualidade de conexão para a transcrição de áudios em tempo real e a precisão variável de modelos de OCR em imagens de baixa resolução. Além disso, a eficácia do acolhimento “empático” por uma IA, embora promissora nos testes, carece de validação com usuárias reais e acompanhamento psicológico.

7.4 TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se a realização de um estudo piloto assistido por profissionais da rede de proteção à mulher, a fim de validar a eficácia do acolhimento em situações reais. Tecnicamente, recomenda-se aprimorar os modelos de reconhecimento de voz e imagem para funcionarem melhor em condições adversas. A expansão para outras plataformas de mensageria amplamente utilizadas, como o *WhatsApp*, também representa um passo natural para aumentar o alcance da ferramenta.

7.5 CONSIDERAÇÕES FINAIS

O MarIA representa um passo inicial, porém firme, na direção de usar tecnologia de ponta para combater uma das violações de direitos humanos mais persistentes. Este trabalho evidenciou que a tecnologia, quando projetada com empatia e responsabilidade, pode ser uma aliada poderosa na proteção e no acolhimento de mulheres, oferecendo um canal de escuta que está sempre disponível.

Referências Bibliográficas

Anderson, Ross J. (2020), *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3^a edição, Wiley. Especialmente os capítulos sobre Criptografia (3), Controle de Acesso (5) e Testes (24).

Aquino, Victor Hugo de Oliveira & Mario Henrique Akihiko da Costa Adaniya (2018), ‘Desenvolvimento e aplicações de chatbot’, *Revista Terra & Cultura: Cadernos De Ensino E Pesquisa* 34(esp.), 56–68.

URL: <http://periodicos.unifil.br/index.php/Revistateste/article/view/297>

Bender, Emily M., Timnit Gebru, Angelina McMillan-Major & Shmargaret Shmitchell (2021), On the dangers of stochastic parrots: Can language models be too big?, em ‘Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)’, Association For Computing Machinery, Nova Iorque, pp. 610–623. Acesso em: 31 maio 2025.

URL: <https://d1.acm.org/doi/epdf/10.1145/3442188.3445922>

Bibault, Jean-Emmanuel, Benjamin Chaix, Arthur Guillemassé & et al. (2019), ‘A chatbot versus physicians to provide information for patients with breast cancer: A randomized controlled trial’, *Journal of Clinical Oncology* 37(15_suppl), 6506–6506.

Brasil (2006), ‘Lei n. 11.340, de 7 de agosto de 2006 - lei maria da penha’. Acesso em: 31 maio 2025.

URL: http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2006/lei/111340.htm

Brasil (2018), ‘Lei n. 13.709, de 14 de agosto de 2018 - lei geral de proteção de dados’. Acesso em: 15 de julho de 2025.

URL: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm

Brown, Tom, Benjamin Mann, Nick Ryder & et al. (2020), ‘Language models are few-shot learners’, *Advances in Neural Information Processing Systems (NeurIPS)* 33.

URL: https://papers.nips.cc/paper_files/paper/2020/file/1457c0d6bfcba4967418bfb8ac142f64a-Paper.pdf

Burr, Christopher, Nello Cristianini & James Ladyman (2020), ‘An analysis of the interaction between intelligent software agents and human users’, *Minds and Machines* 30(2), 201V–220.

Caldeira, Ana Paula dos Santos C. da S. C., Isabela Caldeira, Thallita V. de A. Lima & Juliana C. de O. M. da Silva (2024), ‘Aplicativos móveis de denúncia de violência contra mulheres no brasil: uma análise sobre o desenvolvimento e suas funcionalidades’, *Revista Brasileira de Segurança Pública* 18(1).

URL: <https://revistasusp.mj.gov.br/susp/index.php/revistasusp/article/view/633>

Colin, Galen (2024), ‘Zod: Typescript-first schema validation with static type inference’, Repositório do GitHub. Acesso em: 17 de setembro de 2025.

URL: <https://github.com/colinhacks/zod>

DataSenado (2023), ‘Pesquisa nacional de violência doméstica e familiar contra a mulher – rio grande do norte’. Acesso em: 07 de julho de 2025.

URL: <https://www12.senado.leg.br/institucional/datasenado>

Docker, Inc. (2024), ‘What is docker?’. Acesso em: 17 de setembro de 2025.

URL: <https://www.docker.com/why-docker/>

Few, Stephen (2013), *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*, 2^a edição, Analytics Press. Especialmente o Capítulo 1 (Dashboards: An Introduction) e o Capítulo 3 (The Fundamental Challenges of Dashboard Design).

Fielding, Roy T. (2000), Architectural Styles and the Design of Network-based Software Architectures, Tese de doutorado, University of California, Irvine. Tese de Doutorado.

URL: https://www.ics.ucl.ac.uk/~fielding/pubs/dissertation/fielding_dissertation.pdf

Fórum Brasileiro de Segurança Pública (2024), ‘Anuário brasileiro de segurança pública’, INFOGRÁFICO Anuário Brasileiro de Segurança Pública 2024. Acesso em: 31 maio 2025.

URL: <https://forumseguranca.org.br/anuario-brasileiro-seguranca-publica/>

Google (2024), ‘Dialogflow documentation’. Acesso em: 17 de setembro de 2025.

URL: <https://cloud.google.com/dialogflow/docs>

Governo do Estado de Minas Gerais (2024), ‘Mg mulher - integração e ações de proteção’, <https://www.seguranca.mg.gov.br/index.php/integracao/mg-mulher>. Acesso em: 07 nov. 2025.

Governo do Estado de São Paulo (2023), ‘Sp mulher: São paulo lança aplicativo e abre sala para vítimas de violência’, <https://www.sp.gov.br/sp/canais-comunicacao/noticias/sp-mulher-sao-paulo-lanca-aplicativo-e-abre-sala-para-vitimas-de-violencia/>. Acesso em: 07 nov. 2025.

- Governo do Estado do Paraná (2024), ‘Acionar botão do pânico virtual - vítimas de violência doméstica’, <https://www.semipi.pr.gov.br/servicos/Servicos/Emergencia/Acionar-Botao-do-Panico-virtual-vitimas-de-violencia-domestica-QJoR8b3w>. Acesso em: 07 nov. 2025.
- Governo do Estado do Rio de Janeiro (2024), ‘App rede mulher’, <https://sepm.rj.gov.br/app-rede-mulher/>. Acesso em: 07 nov. 2025.
- Instituto AzMina (2024), ‘Aplicativo PenhaS’. Acesso em: 15 de julho 2025.
URL: <https://www.azmina.com.br/penhas>
- Instituto Metrópole Digital / Governo do Rio Grande do Norte (2023), ‘Salve elas: aplicativo de combate à violência contra a mulher no rn’. Acesso em: 07 de julho de 2025.
URL: <https://imd.ufrn.br/portal/noticias/7194/imd-desenvolve-app-para-combater-violencia-contra-mulheres>
- Jobin, Anna, Marcello Ienca & Effy Vayena (2019), ‘The global landscape of ai ethics guidelines’, *Nature Machine Intelligence* 1(9), 389–399.
- Jurafsky, Daniel & James H. Martin (2025), *Speech and Language Processing*, 3^a edição, Prentice Hall. Versão preliminar online. Especialmente os capítulos 1 (Introduction). Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>.
- MDN Web Docs (2022), ‘Javascript’. Acesso em: 17 de setembro de 2025.
URL: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- Microsoft (2024), ‘Azure bot service documentation’. Acesso em: 17 de setembro de 2025.
URL: <https://docs.microsoft.com/en-us/azure/bot-service/>
- Ministério da Justiça e Segurança Pública (2024), ‘Painéis estatísticos’. Acesso em: 31 maio 2025.
URL: <https://app.powerbi.com/view?r=eyJrIjoiZWUwOWJhYmYtZjAzNi00ZmRkLWJlZmMtODQ3Njd1ZmZjNTZlIiwidCI6In>
- Ministério Público de São Paulo (2021), ‘Aurora: chatbot para apoio a mulheres vítimas de violência’. Acesso em: 07 de julho de 2025.
URL: <https://www.mpsp.mp.br/portal/page/portal/Especial/Aurora>
- OpenJS Foundation (2023), ‘About node.js’. Acesso em: 17 de setembro de 2025.
URL: <https://nodejs.org/en/about/>
- PostgreSQL Global Development Group (2024), ‘About postgresql’. Acesso em: 17 de setembro de 2025.
URL: <https://www.postgresql.org/about/>

Pressman, Roger S. & Bruce R. Maxim (2016), *Engenharia de Software: Uma Abordagem Profissional*, 8^a edição, AMGH Editora, Porto Alegre. A definição de framework é abordada no Capítulo 9: Projeto de Arquitetura.

Richardson, Leonard & Sam Ruby (2007), *RESTful Web Services*, O'Reilly Media, Sebastopol, CA. Especialmente o capítulo 4 (The Resource-Oriented Architecture).

Russell, Stuart & Peter Norvig (2010), *Artificial Intelligence: A Modern Approach*, 3^a edição, Pearson. Especialmente os capítulos introdutórios sobre a definição e escopo da IA.

Secretaria de Comunicação Social do Governo do Rio Grande do Norte (2025), ‘Rn teve três mil medidas protetivas a mais julgadas em 2024’. Acesso em: 31 maio 2025.

URL: <https://saibamais.jor.br/2025/03/rn-teve-tres-mil-medidas-protetivas-a-mais-julgadas-em-2024>

Serban, Iulian V., Ryan Lowe, Peter Henderson, Laurent Charlin & Joelle Pineau (2018), ‘A survey of available corpora and evaluation metrics for building dialogue systems’, *Dialogue & Discourse* 9(1), 1–49.

URL: <https://journals.uic.edu/ojs/index.php/dad/article/view/10733/9501>

SOS Maria da Penha (2024), ‘Sos maria da penha - aplicativo’, <https://sosmariadapenha.com.br/>. Acesso em: 07 nov. 2025.

Tribunal de Justiça do Estado do Rio de Janeiro (2021), ‘Maria da penha virtual’. Acesso em: 15 de julho 2025.

URL: <https://www3.tjrj.jus.br/mariapenhavirtual/>

Woebot Labs Inc. (2017), ‘Woebot: your self-care expert’. Aplicativo de apoio psicológico baseado em chatbot.

URL: <https://woebothealth.com>