

UNIVERSIDADE ESTADUAL DO MARANHÃO – UEMA  
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT  
CURSO DE ENGENHARIA DA COMPUTAÇÃO

**SISTEMA WEB PARA DELEGACIAS COM GERAÇÃO DE IMAGENS  
FOTORREALISTAS BASEADA EM INTELIGENCIA ARTIFICIAL A PARTIR DE  
RELATOS TEXTUAIS**

Hugo Roberto Veras Mendes

São Luís – MA

2026

Hugo Roberto Veras Mendes

**SISTEMA WEB PARA DELEGACIAS COM GERAÇÃO DE IMAGENS  
FOTORREALISTAS BASEADA EM INTELIGENCIA ARTIFICIAL A PARTIR DE  
RELATOS TEXTUAIS**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia da Computação, da Universidade Estadual do Maranhão, como pré-requisito para alcançar o título de Bacharel em Engenharia da Computação.

Orientador: Prof. Dr. Pedro Brandão Neto.

São Luís – MA

2026

Mendes, Hugo Roberto Veras.

Sistema Web para delegacias com geração de imagens fotorrealistas baseadas em Inteligência Artificial a partir de relatos textuais / Hugo Roberto Veras Mendes. - São Luís - MA, 2026.

48 f.

Monografia (Graduação em Engenharia da Computação) - Universidade Estadual do Maranhão, São Luís, 2026.

Orientador: Prof. Dr. Pedro Brandão Neto.

1. Segurança Pública. 2. Inteligência Artificial. 3. Stable Diffusion. I. Título.

CDU: 004:351.78

**Elaborado por Luciana de Araújo - CRB 13/445**

Hugo Roberto Veras Mendes


**SISTEMA WEB PARA DELEGACIAS COM GERAÇÃO DE IMAGENS  
FOTORREALISTAS BASEADA EM INTELIGENCIA ARTIFICIAL A PARTIR DE  
RELATOS TEXTUAIS**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia da Computação, da Universidade Estadual do Maranhão, como pré-requisito para alcançar o título de Bacharel em Engenharia da Computação.


Orientador: Prof. Pedro Brandão Neto.

Aprovada em / /

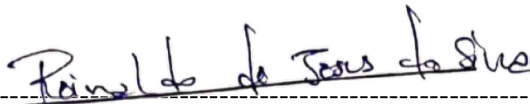
**BANCA EXAMINADORA**

Documento assinado digitalmente  
 **PEDRO BRANDAO NETO**  
Data: 20/03/2026 16:27:15-0300  
Verifique em <https://validar.it.gov.br>

Prof. Me. Pedro Brandão Neto (Orientador)  
Mestre em Engenharia Elétrica  
Universidade Estadual do Maranhão



Prof. Dr. Reinaldo de Jesus da Silva  
Doutor em Informática na Educação  
Universidade Estadual do Maranhão



Prof. Dr. Luís Carlos Costa Fonseca  
Doutor em Informática na Educação  
Universidade Estadual do Maranhão

## **AGRADECIMENTO**

Agradeço este trabalho, primeiramente, aos meus pais, que sempre acreditaram em mim e me ensinaram, com amor e paciência, que a perseverança é a chave para alcançar qualquer sonho. Cada palavra de incentivo, cada gesto de apoio e cada sacrifício feito ao longo da minha jornada foram fundamentais para que eu chegasse até aqui. Este momento é, sem dúvida, reflexo do amor e da força que recebi de vocês.

Agradeço também à minha namorada, que esteve ao meu lado em todos os momentos, oferecendo não apenas companhia, mas também motivação diária para que eu nunca desistisse, mesmo diante dos desafios. Sua compreensão, carinho e incentivo fizeram toda a diferença, tornando essa caminhada mais leve e significativa.

## RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação web voltada para o gerenciamento de delegacias, com foco na integração de inteligência artificial (IA) para geração de imagens fotorrealistas a partir de descrições textuais, visando otimizar investigações criminais. O sistema foi projetado com arquitetura cliente-servidor, utilizando o framework Next.js para o front-end, Flask para o back-end e o modelo Stable Diffusion para a geração de imagens. A metodologia adotada foi aplicada, exploratória e experimental, empregando prototipação incremental para garantir escalabilidade e usabilidade. O levantamento de requisitos identificou lacunas em sistemas de delegacias virtuais, como a ausência de ferramentas visuais avançadas, orientando o desenvolvimento de funcionalidades como criação, gerenciamento e finalização de boletins de ocorrência, autenticação segura via JWT e integração com IA. A modelagem do sistema utilizou UML, com diagramas de caso de uso, classes, sequência, atividades e implantação. Testes de integração e funcionais, realizados com ferramentas como Postman e Cypress, validaram a eficiência e robustez da aplicação. Os resultados demonstraram que o sistema atende aos objetivos, oferecendo uma solução inovadora que agiliza a identificação de suspeitos e melhora a eficiência do registro de ocorrências, com potencial para integração futura com bancos biométricos e adaptação do modelo de IA.

**Palavras-chave:** Segurança Pública, Inteligência Artificial, Stable Diffusion, Sistemas Web, Investigação Criminal.

## *ABSTRACT*

This work presents the development of a web application designed for police station management, focusing on integrating artificial intelligence (AI) to generate photorealistic images from textual descriptions to enhance criminal investigations. Built on a client-server architecture, the system employs Next.js for the front-end, Flask for the back-end, and the Stable Diffusion model for image generation. The methodology combined applied, exploratory, and experimental approaches with incremental prototyping to ensure scalability and usability. Requirements analysis identified gaps in virtual police station systems, such as the lack of advanced visual tools, guiding the development of features like creation, management, and finalization of incident reports, secure JWT authentication, and AI integration. System modeling utilized UML, including use case, class, sequence, activity, and deployment diagrams. Integration and functional tests, conducted with tools like Postman and Cypress, validated the application's efficiency and robustness. The results demonstrate that the system meets its objectives, providing an innovative solution that streamlines suspect identification and improves incident reporting efficiency, with potential for future integration with biometric databases and further AI model adaptation.

**Keywords:** Public Safety, Artificial Intelligence, Stable Diffusion, Web Systems, Criminal Investigation.

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>9</b>
<b>1.1. Objetivos .....</b>	<b>10</b>
<b>1.1.1. Objetivo Geral.....</b>	<b>10</b>
<b>1.1.2. Objetivos específicos .....</b>	<b>10</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>11</b>
<b>2.1. Sistemas Web e Arquitetura Cliente-Servidor.....</b>	<b>11</b>
<b>2.1.1. Sistema WEB .....</b>	<b>11</b>
<b>2.1.2. Arquitetura Cliente – Servidor .....</b>	<b>12</b>
<i>2.1.2.1. Cliente .....</i>	<i>12</i>
<i>2.1.2.2. Servidor .....</i>	<i>14</i>
<b>2.2. Desenvolvimento Front-End com Next.js .....</b>	<b>15</b>
<b>2.2.1. React.....</b>	<b>15</b>
<b>2.2.2. Next.js .....</b>	<b>15</b>
<b>2.3. Desenvolvimento Back - End com Flask.....</b>	<b>16</b>
<b>2.3.1. Python (Flask) .....</b>	<b>16</b>
<b>2.3.2. Microsserviços.....</b>	<b>17</b>
<b>2.4. Redes Neurais e Geração de imagens a partir de texto.....</b>	<b>18</b>
<b>2.4.1. Redes Geradoras Adversariais (GANs) .....</b>	<b>18</b>
<b>2.4.2. Modelos de Difusão (Diffusion Models) .....</b>	<b>19</b>
<b>2.4.3. Stable Diffusion .....</b>	<b>19</b>
<b>2.5. Tecnologias no sistema publico .....</b>	<b>21</b>
<b>2.5.1. Modelo tradicional x atual de registro de ocorrências e avanço tecnológicos na             segurança pública. ....</b>	<b>22</b>
<b>2.5.2. O papel da inteligência artificial na segurança pública .....</b>	<b>23</b>
<b>3. RESUMO DA METODOLOGIA .....</b>	<b>25</b>
<b>3.1. Introdução à Metodologia .....</b>	<b>25</b>
<b>3.2. Levantamento de Requisitos .....</b>	<b>25</b>
<b>3.3. Modelagem do Sistema .....</b>	<b>26</b>
<b>3.4. Prototipação .....</b>	<b>28</b>
<b>3.5. Desenvolvimento do Sistema .....</b>	<b>29</b>
<b>3.5.1. Tecnologias Utilizadas .....</b>	<b>30</b>
<b>3.5.2. Arquitetura do Sistema .....</b>	<b>31</b>
<b>3.5.3. Integração com IA .....</b>	<b>32</b>

<b>3.6. Integração da Inteligência Artificial .....</b>	<b>33</b>
<i>3.6.1. Modelo e Ferramentas Utilizadas .....</i>	<i>33</i>
<i>3.6.2. Processo de Integração .....</i>	<i>34</i>
<i>3.6.3. Ajustes e Personalizações .....</i>	<i>34</i>
<i>3.6.4. Funcionamento no Sistema .....</i>	<i>35</i>
<b>3.7. Testes e Validação .....</b>	<b>35</b>
<b>3.8. Critérios de Avaliação .....</b>	<b>36</b>
<i>3.8.1. Avaliação do Objetivo Geral.....</i>	<i>37</i>
<i>3.8.2. Avaliação dos Objetivos Específicos .....</i>	<i>37</i>
<i>3.8.3. Experiência do Usuário .....</i>	<i>43</i>
<b>3.9. Metodologia.....</b>	<b>43</b>
<i>3.9.1. Pesquisa Aplicada .....</i>	<i>43</i>
<i>3.9.2. Pesquisa Exploratória .....</i>	<i>43</i>
<i>3.9.3. Pesquisa Experimental .....</i>	<i>43</i>
<i>3.9.4. Prototipação Incremental .....</i>	<i>44</i>
<i>3.9.5. Integração Tecnológica .....</i>	<i>44</i>
<i>3.9.6. Eficácia da Metodologia .....</i>	<i>44</i>
<b>4. CONCLUSÃO .....</b>	<b>45</b>
<b>REFERÊNCIAS .....</b>	<b>46</b>

## 1. INTRODUÇÃO

Desde o surgimento do primeiro site da história, amplamente conhecido como "The Project", há mais de 30 anos, temos sido espectadores de uma significativa expansão nos sistemas e dispositivos web. Esse crescimento é impulsionado pela constante evolução tecnológica ao longo dos anos, proporcionando aos usuários experiências cada vez mais práticas e inovadoras.

Os sistemas web não apenas revolucionaram a maneira como lidamos e interagimos na internet, como indivíduos mas também representaram um avanço significativo no âmbito da segurança pública e digital. Isso ficou evidente com o surgimento das delegacias online, cujo propósito é facilitar o acesso do cidadão aos serviços de ocorrência policial.

Essa inovação oferece uma notável conveniência ao registrar uma queixa ou abrir um boletim de ocorrência, destacando cada vez mais a convergência entre tecnologia e serviços públicos. Essa ligação tornou-se quase inseparável à medida que a tecnologia evolui e continua a desempenhar um papel fundamental na modernização e acessibilidade dos processos de segurança pública.

Com a evolução dos dispositivos tecnológicos na segurança pública, destaca-se a crescente utilização das Inteligências Artificiais (IAs) nos processos de investigação e prevenção de crimes, tanto dentro quanto fora da internet. Essas ferramentas não apenas têm a capacidade de identificar padrões, mas também podem gerar imagens de maneira eficaz por meio de treinamento adequado. Essa convergência de tecnologias representa um avanço significativo no fortalecimento das capacidades investigativas e na resposta eficiente aos desafios emergentes na segurança pública.

O propósito deste trabalho é desenvolver uma aplicação web, utilizando ferramentas e linguagens contemporâneas, para o gerenciamento eficiente de uma delegacia. A inovação central desta aplicação reside na capacidade, integrada à inteligência artificial, de gerar uma representação fotorealista a partir da descrição de um suspeito, por meio de um retrato falado, durante o processo de criação de um boletim de ocorrência. Sendo assim, isso resultará em uma solução sofisticada e de fácil acesso, agregando eficiência ao sistema de registro de ocorrências criminais como uma forma de aprimorar algo que já existe e é bem difundido entre a população, trazendo praticidade a investigação criminal.

Dentre os principais conceitos abordados neste trabalho, destacam-se os dispositivos web e a geração de imagens realistas por meio do uso de IAs específicas para esse propósito, como as GANs — redes neurais que competem entre si para a criação de imagens. Além disso, o desenvolvimento deste projeto também faz uso de tecnologias atuais e processos avançados, aplicados à criação de um sistema para delegacias, cujo desenvolvimento será detalhado ao longo deste trabalho.

## **1.1. Objetivos**

### **1.1.1. Objetivo Geral**

Desenvolver uma aplicação web capaz de, a partir de uma descrição de um suspeito, gerar uma imagem realista da aparência do indivíduo. Essa ferramenta visa oferecer suporte às forças policiais durante uma investigação criminal, independente da área de atuação, fornecendo uma valiosa assistência visual na identificação e possível localização de um suspeito de delito, fornecendo uma possível aparência do indivíduo com base nos dados previamente passados pela vítima ou retrato falado através do investigador policial.

### **1.1.2. Objetivos específicos**

- Estudar as melhores práticas relacionadas ao desenvolvimento web;
- Pesquisar boas práticas de arquitetura de microsserviços para dar suporte ao Back-end;
- Desenvolver Front-end utilizando as melhores práticas identificadas;
- Estudar Redes Geradoras Adversárias;
- Propor uma solução que itegre um sistema web a uma Inteligência Artificial.
- Adaptar uma arquitetura de redes neurais para a transformação de texto em imagem.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1. Sistemas Web e Arquitetura Cliente-Servidor

Desde o surgimento do primeiro site da história, amplamente conhecido como "The Project", há mais de 30 anos, nota-se uma significativa expansão nos sistemas e dispositivos web. Os sistemas web, como softwares online, proporcionam a interação necessária para a expansão dos programas mantendo sua função de acesso livre e dinamismo entre os usuários.

#### 2.1.1. Sistema WEB

Os sistemas web vieram para renovar o que chamamos de sistemas desktops, que assim como os sistemas web, armazenam dados e fornece através da pesquisa as informações solicitadas pelos usuários. Porém, os sistemas desktops necessitavam de download em um computador físico o que dificultava seus acessos e atualizações, como reforça Silva (2017) *“a principal diferença entre sistemas web e desktop está na arquitetura de execução e na forma de distribuição e atualização das aplicações”*. Os sistemas web fornecem a mesma funcionalidade de forma online onde quaisquer usuários podem acessar dados de um sistema através da internet, isso substitui por exemplo, a ideia de um diário de notas de um professor onde ele armazenava os dados dos alunos de forma escrita através de um caderno de controle, hoje, o professor pode facilmente abrir o sistema da escola em qualquer computador, em qualquer horário, em qualquer lugar e através da internet ele pode incluir notas e divulgá-las aos seus alunos de forma automática e instantânea.

Os sistemas web são hospedeiros virtuais em rede que permitem interações de compartilhamento de informações entre usuários e serviços, cujas principais características estão no seu acesso livre global, armazenamento em servidores online e escaláveis, eficiência operacional e atualizações centralizadas, além da segurança dos dados. De acordo com Dias (2014), sistemas web são aplicações que são acessíveis à maioria dos usuários assim como, são usáveis, aumentando a satisfação das pessoas durante a interação com o computador, desenvolvendo um conforto aos usuários, lidando com aumento de tráfego e número de pessoas.

Há etapas essenciais para o desenvolvimento de um sistema web que garante a partir da sua geração uma forma funcional e de qualidade de gestão de dados, estas que são: Identificar os requisitos e necessidades dos usuários, Design do sistema, escolha da linguagem de programação, testes, implantação e manutenção/atualização do sistema. Com isso, um

modelo de comunicação entre computadores no qual nomeamos como Arquitetura Cliente – Servidor, é utilizado para construção do sistema no qual podemos considerar o cliente como a interface que o usuário manipulará (Front – End), onde os dados estarão armazenados e dispostos em um banco de dados, que será acessado através de uma comunicação em linguagem de programação que buscará os resultados (API) que serão validados (Back-End) e retornados, sendo que, o servidor armazena todo o sistema de forma online.

### 2.1.2. Arquitetura Cliente – Servidor

Quando lembramos de arquitetura podemos recordar de construção, e o significado não se torna diferente quando estamos na engenharia da computação. Arquitetura cliente – servidor é um modelo de construção amplamente utilizado para desenvolvimento de sistemas distribuídos, onde de acordo com Tanenbaum e Van steen (2017) *“um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”*. Nessa construção temos dois tipos de pilares: Cliente e o Servidor.

#### 2.1.2.1. Cliente

Em meio à construção de um sistema web, o cliente é o requisitante dos dados, ou seja, a interface ou navegador que solicita informações ao servidor. Porém, para a determinação do design e da interatividade da interface, utilizamos linguagens como HTML, CSS e JavaScript, sendo que essas são responsáveis por estruturar o conteúdo, definir o estilo e o dinamismo da aplicação, respectivamente.

#### ➤ HTML

Linguagem de Marcação de Hipertexto, ou HTML, é uma linguagem utilizada para definir os padrões de formatação e os elementos da estrutura de um sistema web, sendo eles textos, marcações, imagens e formulários. Criado por Tim Berners-Lee, o HTML passou por diversas atualizações para alcançar o nível de design, interatividade e acessibilidade existente atualmente, mantendo o objetivo de seu criador, conforme mencionado no livro *Weaving the Web* (1999): *“O princípio primário por trás da independência de dispositivo e da acessibilidade é a separação entre forma e conteúdo.”*

Usualmente, o HTML utiliza códigos para estruturar as páginas e os elementos dispostos nelas, como parágrafos, tabelas e imagens. As tags transformam uma palavra ou imagem em um hiperlink e podem ser de abertura ou de fechamento

A tag de abertura consiste no código utilizado para iniciar o conteúdo que se deseja

escrever (<p>), e a tag de fechamento é o código responsável por finalizá-lo, sendo semelhante à tag de abertura, porém com a adição de uma barra (</p>).

### ➤ CSS

Folhas de Estilo em Cascata, ou CSS, trabalham em conjunto com o HTML para estilizar o front-end de um sistema web, tornando tanto o trabalho do desenvolvedor quanto a interação das pessoas com as interfaces mais prática e dinâmica. Ou seja, o CSS define como os elementos do HTML serão dispostos na tela, bem como características visuais como tipo de fonte, cores e formas de exibição, conforme destaca Condo (2004),

“O CSS permite que os produtores da Web mantenham as informações sobre aparência e apresentação separadas do conteúdo e da estrutura dos documentos da Web. Essa separação simplifica a manutenção de sites e reduz a redundância das informações de apresentação.”

O CSS, teoricamente, é a parte estética de um sistema web e pode ser aplicado de três formas: estilo interno, externo ou online. O estilo interno é carregado toda vez que o site é atualizado e não permite a reutilização do mesmo estilo em todas as páginas do sistema, pois está contido em uma única página, sendo inserido diretamente no próprio HTML, geralmente dentro de uma tag como <style>.

Já o estilo externo, como o próprio nome indica, é carregado a partir de um arquivo .css, o que permite sua aplicação em qualquer página desejada, promovendo um visual desejável e um bom desempenho de carregamento do sistema, pois é armazenado em cache pelo navegador.

Por outro lado, o estilo online é aplicado a elementos específicos, e suas tags, assim como no estilo interno, estão contidas dentro do HTML. No entanto, sua manutenção é mais delicada e não separa o conteúdo da apresentação.

### ➤ JavaScript

ECMAScript ou hoje conhecida JavaScript é uma linguagem de programação que tem interação com HTML e CSS na função de dinamismo e comunicação entre o front end e a API.

Após o licenciamento pela Oracle e sua utilização inédita pela Mozilla, a linguagem JavaScript passou ser mais utilizada no desenvolvimento de um sistema web, como reforça Flanagan (2011),

“A ampla maioria dos sistemas modernos usa JavaScript e todos os navegadores modernos – em computadores de mesa, consoles de jogos, tablets e smartphones – incluem, interpretadores JavaScript, tornando-se a linguagem de programação mais onipotente da história.”

Sua popularidade não é somente entre os programadores; os usuários de sites que utilizam essa linguagem percebem a diferença, devido aos elementos da interface dos usuários reagirem em tempo real, com cliques, digitação e atualizações, sem a necessidade de recarregar a página da web. Isso porque o JavaScript tem como um dos principais recursos o DOM (Document Object Model), além de frameworks como React.js e a extensão Next.js, que transformam a estrutura em forma de árvores, permitindo que o JavaScript faça modificações diretamente no HTML, a partir do navegador, potencializando a criação de interfaces dinâmicas, modernas e otimizadas.

#### 2.1.2.2. Servidor

Servidor fornece serviços ao cliente em uma rede, onde é feita operações de sistemas distribuídos. Segundo Tanenbaum e Steen (2007), “*a arquitetura cliente-servidor é a base para a maioria dos sistemas modernos, onde o servidor gerencia recursos e responde a requisições de forma eficiente.*” Assim operando e gerenciando de forma contínua as informações solicitadas pelo usuário garantindo a disponibilidade do sistema.

Por conseguinte, temos vários tipos de servidores, nos quais são utilizados de acordo com a necessidade e o papel que eles irão desempenhar; como, por exemplo, servidores de bancos de dados, servidores de arquivos e servidores web, este último sendo um software que gerencia requisições e devolve respostas ao cliente (navegador).

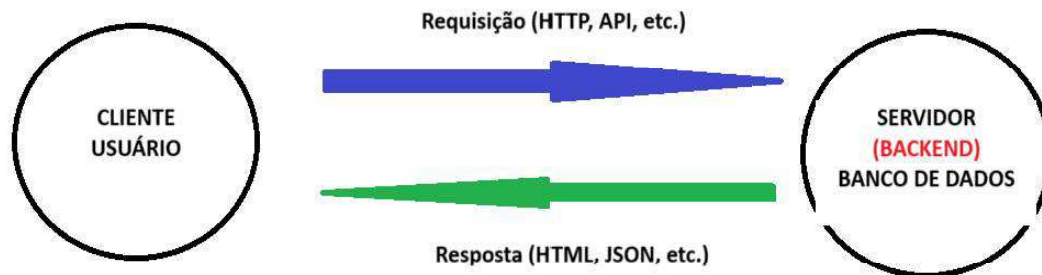
O servidor web hospeda sites e processa requisições HTTP/HTTPS a partir das solicitações feitas pela API. Ou seja, além de armazenar o site, ele processa protocolos de comunicação a fim de assegurar a disponibilidade, escalabilidade, segurança, desempenho e gerenciamento do sistema web.

Desse modo, o protocolo Hypertext Transfer Protocol, ou simplesmente HTTP, se comunica com o servidor, que devolve as respostas em linguagem HTML. Vale ressaltar que o servidor web é um servidor sem estado, o que indica que ele não armazena os dados solicitados; apenas recebe e responde aos pedidos, como mencionado por Silva (2009), “*A confiabilidade das informações em uma transmissão HTTP e feita com o protocolo TCP (Transmission Control Protocol), apesar disso, HTTP é um protocolo sem estado, ou seja, cada comunicação é tratada*

*independentemente.*”

Com isso, fica assim estruturado a arquitetura cliente – servidor

**Figura 1:** Diagrama Arquitetura cliente - servidor



**Fonte:** Elaboração própria

## 2.2. Desenvolvimento Front-End com Next.js

Com a evolução das tecnologias web, as interfaces dos sistemas tornaram-se cada vez mais dinâmicas e interativas. Graças ao desenvolvimento do front-end, o qual abrange a criação da interface desses sistemas, os usuários se tornam mais receptivos e responsivos, pois a agilidade de resposta e a segurança proporcionam maior conforto ao navegar na rede.

Nesse contexto, a evolução e utilização de frameworks para otimização trouxe modernidade e praticidade aos desenvolvedores. Como exemplo, o Next.js é um framework baseado no React, uma ferramenta poderosa para aplicações web. Nela, se usam componentes do React para criar interfaces de usuário, enquanto o Next.js fornece recursos e otimizações adicionais, como mencionado por Next.js (2025).

### 2.2.1. React

React é uma biblioteca JavaScript que permite combinações de componentes para melhor disposição dos atalhos da interface do usuário. Segundo o documento oficial do React, a interface do usuário é construída a partir de pequenas unidades e o React permite combiná-los de forma encaixáveis e reutilizáveis. *O React também pode renderizar no servidor usando o Node e alimentar aplicativos móveis usando o React Native.* (REACT, 2025)

O diferencial do React é a utilização do Virtual DOM, é um conceito de representação virtual de uma interface que é mantida na memória e sincronizada com o DOM “real” por uma biblioteca. (REACT, 2025)

### 2.2.2. Next.js

Um framework é uma forma simples de organizar e estruturar linguagens de programação, sendo o Next.js um framework baseado no react, ele auxilia na função de

organização e fornecimento de recursos adicionais no qual expande os códigos do react para aplicações modernas como, páginas estáticas e roteamento automático, permitindo ao usuário uma experiência otimizada e fácil de usar na sua interface.

Ele também configura automaticamente ferramentas de nível inferior, como empacotadores e compiladores. Em vez disso, você pode se concentrar em criar seu produto e enviar rapidamente. Seja você um desenvolvedor individual ou parte de uma equipe maior, Next.js pode ajudá-lo a criar aplicativos React interativos, dinâmicos e rápidos. (NEXT.js , 2025)

O Next.js utiliza dois roteadores sendo eles de aplicativos e de páginas onde, o primeiro é o mais atual e oferece novos recursos do React como componentes do servidor (NEXT.js 2025) e o segundo o roteador original que está sendo aprimorado.

Contudo, junto com o SSR ou Server - Side Rendering e Static Generation (SSG), o Next.js facilita a renderização do lado do servidor permitindo uma tela leve e páginas que carregam instantaneamente, além de criar rotas de API sem back - end separado, rotas de arquivos e uma performance otimizada por padrão.

### **2.3. Desenvolvimento Back - End com Flask**

Back-end é a parte do sistema web que roda no servidor e não é visível ao usuário, ou seja, o “cérebro” por trás da aplicação cuja responsabilidade é de processar dados, gerenciar banco de dados, executar regras, gerenciar autenticação/segurança e responder ao front - end com dados prontos para exibição em tela. Com isso, é comum encontrarmos linguagens como JavaScript, Python, PHP etc. assim como banco de dados: MySQL ou SQL server, APIs com REST na sua estrutura básica.

#### **2.3.1. Python (Flask)**

Criado no início dos anos 90 por Guido Van Rossum e seus companheiros (Python Software Foundation, 2025), Python se tornou hoje uma das linguagens mais usadas por programadores devido à sua simplicidade, legibilidade, ampla bibliotecas e execução em multiplataformas.

Tendo isso, o Python destaca-se no âmbito do back-end pelo suporte a frameworks como Flask e Django, que constroem APIs REST de forma organizada e segura, oferecendo também um suporte para conexões HTTP e gerenciamento de rotas que são fundamentais para disponibilidade nos dados via sistema web, com compatibilidade com vários bancos de dados como MySQL, PostgreSQL e SQLServer. Python se propõe a trazer mais comodidade na hora de programar, destacando - se diante das demais linguagens, principalmente por ser otimizado, o que indica ter menos códigos na tela do desenvolvedor.

Contudo, o conjunto Python e framework Flask desenvolve um back-end mais aderente às boas práticas da engenharia sendo o Flask responsável por trazer recursos para criação de aplicações modulares, leves e flexíveis, sendo que, no decorrer do desenvolvimento do back-end, podemos adicionar apenas componentes relevantes como sistema de autenticação e banco de dados.

De acordo com a documentação oficial do Flask, (Pallets Project, 2025), utiliza - se o padrão Web Server Gateway Interface ou simplesmente, WSGI, e foi projetado para um início rápido e fácil, com capacidades de escalar aplicações complexas e com aderência em diversos servidores web, proporcionando uma liberdade idealizada para micros serviços e sistemas web.

### 2.3.2. Microsserviços

Sabendo da arquitetura cliente – servidor, entendemos que um sistema web é desenvolvido a partir dos pilares do front–end (interface), back–end (cérebro por trás) e um servidor, no qual é armazenado o sistema. Dados esses critérios, há duas abordagens de como serão modulados essas linguagens durante o processo de criação, sendo eles a abordagem clássica Monolítica e a abordagem contemporânea de Microsserviços.

A abordagem Monolítica é baseada em uma junção de todas as funcionalidade de um sistema dentro de um único código, cuja interface, requisitos e acessos aos dados, estão interligados, o que implica dizer que, a cada desenvolvimento ou manutenção feita em um sistema que utiliza essa abordagem pode se tornar cansativo e trabalhoso pois, os códigos se tornam cada vez maiores e mais complexos o que não favorece à grandes empresas por exemplo.

A abordagem mais utilizada e contemporânea é a dos microsserviços, nela podemos contar com fragmentos das funcionalidades do sistema, onde cada um é responsável por sua função, porém, fazem parte de um todo que trabalha paralelamente no desenvolvimento do sistema, como menciona Fowler, 2017

“Se tivermos várias tarefas, poderemos processá-las todas ao mesmo tempo enviando – as para um grupo de workers que possam processá-las em paralelo. Se precisarmos processar mais tarefas, poderemos facilmente dimensionar a aplicação de acordo com a demanda, acrescentando workes adicionais para processar as novas tarefas sem afetar a eficiência de nosso sistema.”

Um microsserviço interage entre si usando RPC (Remote Procedure Calls) ou por

endpoints de API de outros microsserviços ou seja, ele utiliza protocolos específicos e enviará pela rede alguns dados em formato padronizado para outro serviço (Fowler, 2017). O mais comum meio de comunicação entre serviços é o HTTP+REST/THRIFT, este que usa hipertexto para enviar solicitações e receber endpoints específicos no formato REST, principalmente por ser fácil configurar ele se torna confiável e estável sendo sua grande desvantagem a necessidade de ser síncrono.

A usabilidade da abordagem de microsserviços propõem maior facilidade de manutenção, rápida entrega e maior tolerância de falhas por outro lado, ele impõe desafios no gerenciamento, necessidade de automação e maior comunicação entre serviços.

#### **2.4. Redes Neurais e Geração de imagens a partir de texto.**

Estamos vivenciando uma era onde as inteligências artificiais estão dominando as plataformas digitais. Isso se dá pelo avanço das tecnologias baseadas no conceito das redes neurais o qual tem sido idealizado a partir do próprio entendimento do cérebro humano. Um neurônio humano é capaz de processar uma informação mais rápido que qualquer computador já existente, o que coloca nossos neurônios como o maior processador existente. As redes neurais são modelos computacionais capazes de aprender padrões a partir de dados dispostos e disponíveis como, por exemplo, a capacidade da geração de imagens computacionais de acordo com uma descrição em texto.

O procedimento utilizado para realizar a aprendizagem de uma rede neural é dominado de algoritmo de aprendizagem onde, sua função é modificar os pesos sinápticos da rede de forma ordenada a fim de alcançar seu objetivo (ENGEL, 2007), dessa forma que a rede neural se fortalece se tornando cada vez mais robusta e sofisticada.

Uma rede neural é formada por várias camadas sendo a primeira a camada de entrada ou input, onde ela recebe os dados que serão trabalhados, posteriormente temos camadas ocultas, ou *hidden layers*, nos quais ocorrem todos os cálculos, dos mais simples aos mais complexos. Por fim, temos a camada de saída ou output, no qual é gerado o resultado, caso o resultado não seja satisfatório, a rede neural é retroalimentada e seus dados são reavaliados para que seja treinada novamente, a fim de cumprir as vantagens de versatilidade, capacidade de aprendizagem, generalização, processamento paralelo, adaptação e personalização.

##### **2.4.1. Redes Geradoras Adversariais (GANs)**

As redes Geradora Adversarias ou simplesmente GANs, foi desenvolvida por Yann LeCun (GOMES, J.C., BRUNO, R. D.,2023) dentro do aprendizado sobre máquinas generativas que reside na estrutura composta por duas redes neurais artificiais de formas opostas

que são: Gerador e Discriminador, eles que são responsáveis por criar amostras sintéticas a partir de uma entrada aleatória e tem uma função de distinguir se uma determinada amostra é real (proveniente do treinamento do gerador), respectivamente. *A otimização dos GANs termina em um ponto mínimo de sela do gerador, ou seja, o ponto de sela é o mínimo em relação ao gerador e o máximo em relação ao discriminador, buscando a otimização para atingir o equilíbrio de Nash.* (GOMES, J.C., BRUNO, R. D.,2023)

Desde a sua criação as GANs têm sido notáveis em diversas áreas como a geração de rostos humanos realistas, imagens fotorrealistas e até mesmo com os vídeos manipulados digitalmente que geram fakes News. Sendo assim, as GANs vêm para renovação e evolução das tecnologias, aumentando a visão computacional, mídias e ciências aplicadas, como transformar ações que antes eram impensáveis e computacionalmente complexas em algo que é de fácil acesso.

#### 2.4.2. Modelos de Difusão (Diffusion Models)

O modelo de difusão é uma classe de modelos generativos que tem como base dois estágios, sendo eles: a difusão e remoção de ruído. A difusão ou degradação progressiva, é um modelo que adiciona ruído de forma gradual a uma imagem até que ela fique pura. Em seguida, temos a remoção do ruído que trata de aprender a reverter o processo anterior, o que implica que, irá reconstruir a imagem realista a partir do ruído mencionado por Andrade e Oliveira (2024) são modelos de aprendizado de máquina em que o ruído é introduzido e depois retirado do original.

Esse processo utiliza cálculos complexos e de princípios da física estatística, mas permite imagens mais estáveis e de qualidade comparados aos modelos GANs que, apresentavam instabilidade no treinamentos, colapsos (criação da mesma imagem) e dificuldade de controle do que estava sendo gerado no entanto, os modelos de difusão são mais realistas e têm fidelidade de geração.

#### 2.4.3. Stable Diffusion

Stable Diffusion é um modelo de difusão que se destacou por gerar imagens a partir de descrições textuais, chamado de *text-to-image*, onde temos uma ferramenta de alta qualidade, aberto e acessível, como destaca no documento oficial, 2025

Stable Diffusion Online é um gerador de imagens gratuito de Inteligência Artificial que cria com eficiência imagens de alta qualidade a partir de prompts de texto simples. Ele foi projetado para designers, artistas e criativos que precisam de criação de imagens rápida e fácil.

O Stable Diffusion é uma ferramenta poderosa para criação de imagens a partir de texto, com uma interface amigável, sem custos, com resultados de alta qualidade, flexível e personalizável. Esse modelo é baseado em difusão latente, ou seja, ele transforma imagens para um espaço latente mais eficiente, reduzindo o custo operacional, sem comprometer a qualidade do produto. Para isso, utiliza codificações como CLIP ou T5 para interpretar as linguagens naturais. Além disso, por ser de código aberto, torna-se mais popular entre os desenvolvedores. O Stable Diffusion utiliza tecnologias de difusão que têm aplicações na criação de imagens artísticas, design de produtos, conteúdos para jogos e filmes, marketing e publicidades automatizados.

**Figura 2:** Resultado de uma descrição usando Stable Diffusion de um gato na sacada



**Fonte:** Site oficial do Stable Diffusion

**Figura 3:** Resultado de uma descrição usando Stable Diffusion de um menino



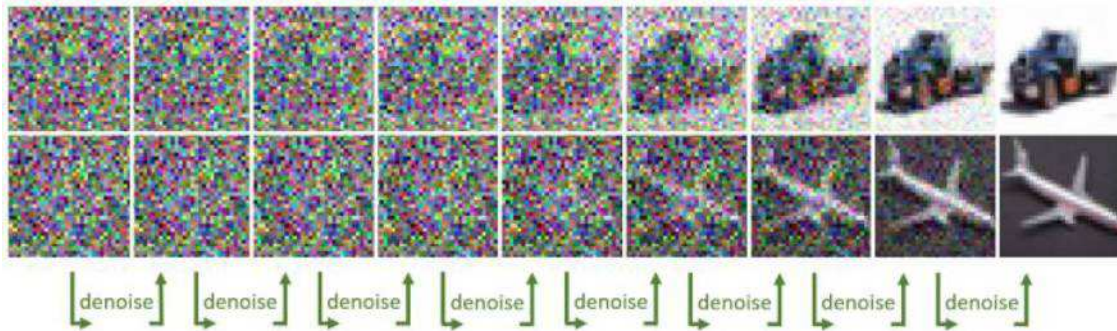
**Fonte:** Site oficial do Stable Diffusion

Com isso, fazendo a comparativa com as GANs, temos menor estabilidade de treinamento, falhas de imagens, risco de colapso, controle de entrada e texto limitado. Em contrapartida, os modelos de difusão como stable, é mais interpretável, tem maior estabilidade, alta qualidade, com maior diversidade de amostras e alto controle de entrada, tornando estes modelos os favoritos dentro do mundo dos desenvolvedores.

O processo de geração da imagem inicia-se com a descrição da imagem que deseja obter. Essa mensagem é decodificada e transformada por meio do modelo CLIP (Pré – treinamento contrativos de texto e imagem) em vetores numéricos e posteriormente o Autoencoder Variacional ou VAE, codifica o que será a imagem. Assim explana Duarte, 2025

Três parâmetros controlam a dinâmica dessa geração a seed (semente aleatória que determina o ruído inicial e garante reprodutibilidade), o número de steps (iterações de remoção de ruído) e a escala CFG (Classifier – Free Guidance), que define quão fortemente o modelo deve seguir a descrição textual em vez de sua distribuição interna de imagens

**Figura 4:** Ilustração do processo de difusão reversa



Fonte: Duarte, 2025

## 2.5. Tecnologias no sistema publico

Apesar da divulgação do Ministério da Justiça e Segurança mostrar redução de casos de homicídios em 2025, as taxas de feminicídios, por exemplo, vem aumentando a cada dia no Brasil, com média (dados de 2024) 196 casos por dia (Ministério da Justiça e Segurança Pública, 2025). O uso de tecnologias dentro do órgão público, é essencial para identificação e posteriormente prisão dos suspeitos. Os registros de boletim de ocorrências eram feitos manualmente, o que apresentava um atraso nos avanços de investigações, agilidade dos atendimentos, na identificação e prisão do acusado. A informatização de delegacias e o uso de tecnologias avançadas representam uma mudança estrutural cuja forma de como os crimes são registrados, investigados sejam eficientes, transparente e ágeis.

[...] os dados que as tecnologias digitais e as respectivas bases (de dados) recolhem e arquivam são, cada vez mais, da ordem da imagem, com destaque para as imagens dos nossos rostos; e aquelas tecnologias têm vindo a aperfeiçoar cada vez mais, ao longo do tempo, a sua capacidade de processamento dessas imagens. (Jornals OpenEdition, 2022)

Sistemas como SINESP (Sistema Nacional de informações de segurança pública) desenvolvido pelo próprio Ministério da Justiça, consolidam uma infraestrutura nacional necessária onde, podemos a partir de um sistema web, coletar, obter e inserir dados de pessoas desaparecidas, ou alegar roubos de veículos, furtos etc. O uso de IAs pelas delegacias tem

aumentado, principalmente no desenvolvimento de retratos falados ou até mesmo na simulação de retratos atualizados por idade de pessoas desaparecidas permitindo que a prisão ou até mesmo o encontro de crianças que já estão adultas sejam solucionados.

#### 2.5.1. Modelo tradicional x atual de registro de ocorrências e avanço tecnológicos na segurança pública.

O modelo tradicional de registro de ocorrências ou comumente chamado B.O. apresenta uma série de limitações como: demora no registro e arquivamento físico, recuperação e cruzamento de dados, assim como, análises de padrões criminais. A dependência de registros escritos também gerava obstáculo para a comunicação entre diferentes unidades policiais o que comprometia com as investigações especiais. Com isso as filas nas delegacias e os atrasos de atendimento cresciam cada vez mais, trazendo transtorno há quem iria fazer alguma denúncia, como mencionado por Silva e Limeira, 2023 *“Com o crescimento exponencial do armazenamento de informações e o avanço das técnicas de análise de dados, as forças policiais agora têm acesso a uma quantidade impressionante de dados que podem ser usados para prever e prevenir crimes.”*

Sendo assim, a chegada da era da tecnologia o registro e consultas de boletins de ocorrência tornaram –se mais rápidos e organizados. Ferramentas que utilizam banco de dados para armazenamento permitem grandes quantidades de informações com acessos fáceis e ágeis. Além da utilização para armazenamento de informações, outras tecnologias que mantem um sistema integrado possibilitam cruzamento de dados e identificação de padrões, assim como emissões de alertas em tempo real.

Uma das principais áreas em que as novas tecnologias têm revolucionado a segurança pública é a coleta e análise de dados. Com o crescimento exponencial do armazenamento de informações e o avanço das técnicas de análise de dados, as forças policiais agora têm acesso a uma quantidade impressionante de dados que podem ser usados para prever e prevenir crimes. (SILVA E LIMEIRA, 2023)

Atendimentos remotos é uma realidade que favorece e reduz drasticamente as filas que antes eram um empecilho nas delegacias. Denúncias básicas podem hoje serem registradas através de sistemas web que são integrados ao banco de dados da polícia, que chegam diretamente aos delegados prontos para serem investigados. Assim como a utilização de softwares e hardwares de vigilância potencializam a investigação e identificação de possíveis locais onde pessoas procuradas passaram, bem como a utilização de inteligência artificial para a partir da descrição podermos desenvolver retratos falados.

A utilização de câmeras de vigilância também está se tornando cada vez mais comum nas cidades. A tecnologia de reconhecimento facial, por exemplo, pode auxiliar na identificação de suspeitos e na localização de pessoas desaparecidas. (SILVA E LIMEIRA, 2023)

### 2.5.2. O papel da inteligência artificial na segurança pública

Com cálculos cada vez mais precisos e softwares mais avançados, as inteligências artificiais ou IAs, tem desempenhado um papel crescente na segurança pública. Os algoritmos de aprendizado de máquinas, como o já mencionado aqui Stable Diffusion, estão sendo utilizados para detectar suspeitos e na identificação de crianças desaparecidas a partir da idade atual da criança.

Em síntese, a aplicação da inteligência artificial na segurança pública não apenas aprimora a eficiência operacional, mas também redefine as estratégias de resposta e prevenção, moldando um cenário mais adaptável e proativo diante dos desafios emergentes. (Scientifique, 2023)

Além disso, as IAs promovem históricos e prevê áreas de maior risco e analisando grandes volumes de informações textuais para investigações. Nas redes sociais, as IAs são utilizadas pela polícia para monitoramento e combate à fakes News, além de detecção automática de comportamento suspeitos. Porém, para utilização dessa tecnologia exige uma regulamentação adequada para garantir os direitos e eficiências, então foi desenvolvido a “Lei de proteção de dados pessoais” ou LGPD, em 2018 para regulamentar o manejo de informações pessoais e estabelecer diretrizes claras para as empresas e instituições, como mencionado na revista Pro Lege Vigilanda, 2023 por Silva e Limeira

A LGPD marca um importante avanço no âmbito legal do país, proporcionando aos cidadãos um maior controle sobre suas informações e assegurando que seus dados pessoais sejam processados de maneira transparente e segura.

Com isso, as IAs são tecnologias de alta performance e de ponta que permitem circular por todos os âmbitos das delegacias ou agências de segurança, tendo sua própria lei de proteção de dados que proporciona a segurança dentro das profundezas da internet para que possam assegurar os direitos humanos e do desenvolvimento de sistemas mais avançados.

Sua vantagem dentro do âmbito de segurança pública aplica-se desde o monitoramento de câmeras, organizações e fornecimento de dados, projeção de perfis de suspeitos e até mesmo na busca de pessoas desaparecidas.

Tal tecnologia possibilita a utilização em diversos

segmentos da SP. Pode ser utilizada mediante a necessidade da identificação de autores, vítimas ou testemunhas relacionadas a algum fato criminoso, além de poder auxiliar na busca de pessoas desaparecidas. Outro uso é para o monitoramento e controle de acesso, como, por exemplo, a linha BIO-T da Intelbras, lançada em 2022. (VIERA. J. C, 2023)

### **3. RESUMO DA METODOLOGIA**

#### **3.1. Introdução à Metodologia**

Este trabalho adota uma abordagem de pesquisa aplicada, exploratória e experimental, utilizando o modelo de prototipação incremental para o desenvolvimento de uma aplicação web voltada ao aprimoramento do processo investigativo em delegacias. A abordagem aplicada reflete o objetivo de criar uma solução prática que integra inteligência artificial (IA) para gerar imagens fotorrealistas a partir de descrições textuais ou retratos falados, visando otimizar a identificação de suspeitos. O caráter exploratório foi empregado na pesquisa de melhores práticas em desenvolvimento web, arquiteturas de microsserviços e modelos de IA, como o Stable Diffusion. A abordagem experimental envolveu a criação, teste e validação de protótipos funcionais, com iterações para garantir a robustez da solução. O modelo de prototipação incremental permitiu entregas parciais em ciclos curtos, facilitando ajustes com base em validações contínuas e minimizando riscos de falhas estruturais.

A escolha dessa abordagem justifica-se pela necessidade de desenvolver um sistema inovador com integração de tecnologias complexas (IA, microsserviços, autenticação segura), exigindo flexibilidade para ajustes durante o desenvolvimento. A prototipação incremental foi essencial para incorporar feedback contínuo e garantir a conformidade com os requisitos levantados, enquanto a natureza aplicada e exploratória permitiu alinhar o projeto às demandas reais das delegacias e às inovações tecnológicas disponíveis.

#### **3.2. Levantamento de Requisitos**

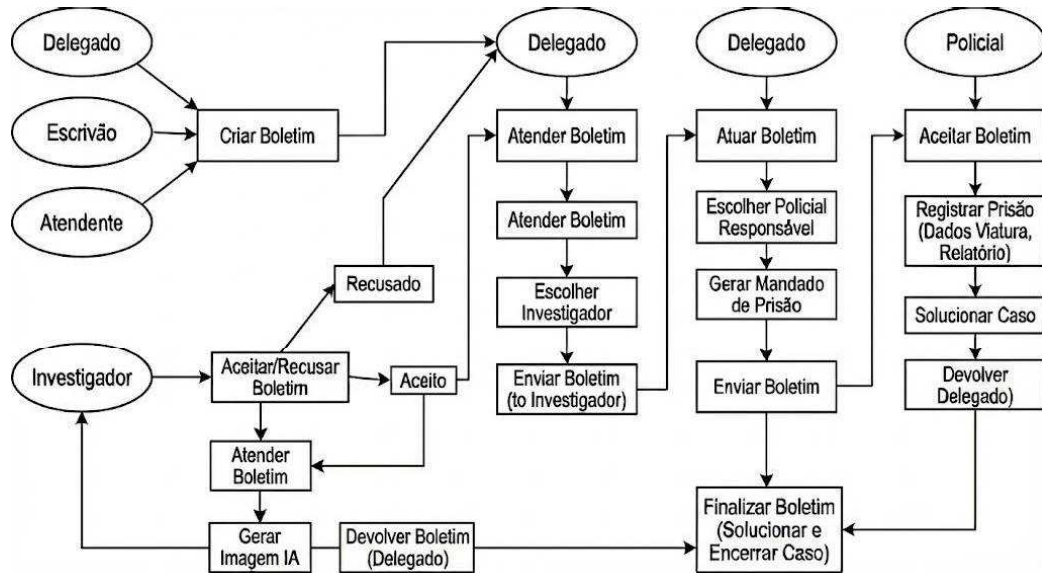
O levantamento de requisitos foi realizado por meio de pesquisa documental, análise comparativa e consulta à literatura técnica. Foram analisados sistemas de delegacias virtuais em estados brasileiros, como Maranhão e Piauí, para identificar funcionalidades essenciais, como criação, gerenciamento e finalização de boletins de ocorrência, autenticação de usuários e integração com IA. Lacunas foram identificadas, como a ausência de ferramentas para geração de imagens realistas de suspeitos, o que orientou o foco do projeto. A pesquisa documental incluiu relatórios técnicos e normativas policiais, enquanto a análise comparativa envolveu benchmarking com sistemas existentes para mapear boas práticas e pontos de melhoria. A consulta à literatura técnica abrangeu desenvolvimento web (ex.: frameworks como Flask e Next.js), arquiteturas de microsserviços e modelos de IA (ex.: Stable Diffusion). Essa abordagem garantiu que os requisitos fossem claros, alinhados às necessidades práticas e tecnicamente viáveis.

### 3.3. Modelagem do Sistema

A modelagem do sistema foi realizada com a Unified Modeling Language (UML) para representar a estrutura e o comportamento da aplicação. Os diagramas elaborados foram:

- Diagrama de Caso de Uso: Representa as interações entre os atores (Delegado, Escrivão, Investigador, Policial, Atendente, Usuário Comum) e as funcionalidades do sistema, como criar boletim, atender boletim, designar investigador, aceitar/recusar boletim, atualizar boletim, gerar imagem com IA, devolver boletim, autuar suspeito e finalizar boletim.

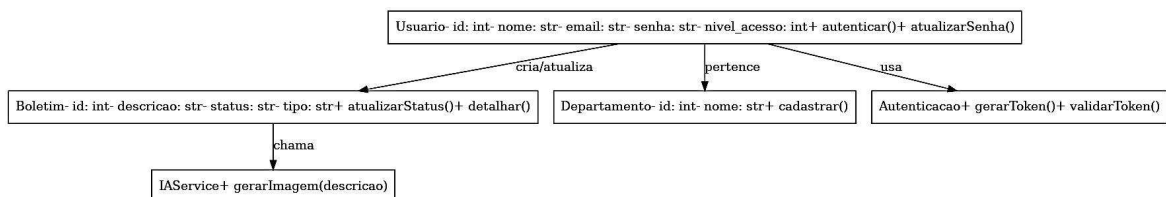
Figura 5: Diagrama de uso do sistema



Fonte: Elaboração própria

- Diagrama de Classes: Detalha a estrutura estática, incluindo classes como Usuario (atributos: id, nome, email, senha, nivelAcesso, departamentoId), Boletim (atributos: id, status, tipoBoletim, descricao, idInvestigador, idDelegado, dataCriacao), Departamento (atributos: id, nome, local) e IAService (atributos: modeloIA, entradaTexto, saidaImagem). Define relacionamentos, como Usuario 1:N Boletim e Boletim 1:1 IAService.

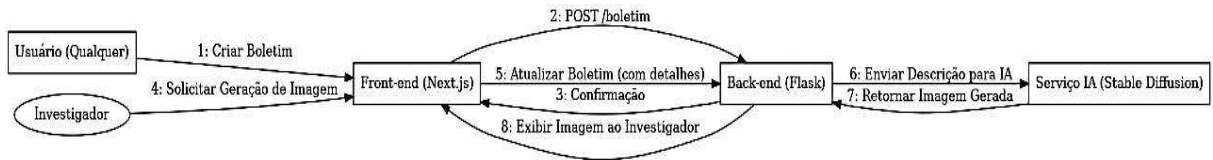
Figura 6: Diagrama de classes do sistema



Fonte: Elaboração Própria

- Diagrama de Sequência: Ilustra a comunicação entre objetos durante o ciclo de vida de um boletim, desde a criação até a finalização, destacando interações entre front-end, back-end, banco de dados e IA.

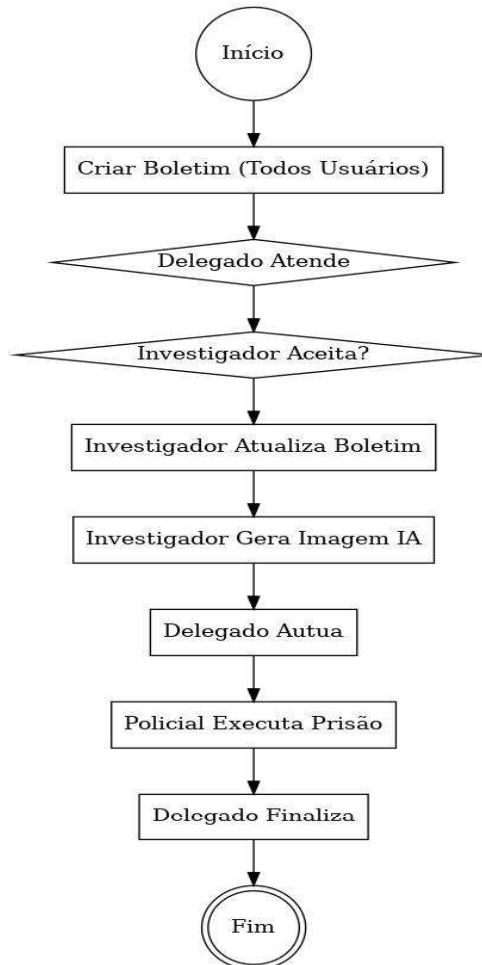
Figura 7: Diagrama de sequência do sistema



Fonte: Elaboração Própria

- Diagrama de Atividades: Descreve o fluxo de operações e decisões, como criação, decisão do Delegado (atender?), decisão do Investigador (aceitar?), atualização, devolução, autuação, prisão, solução e finalização.

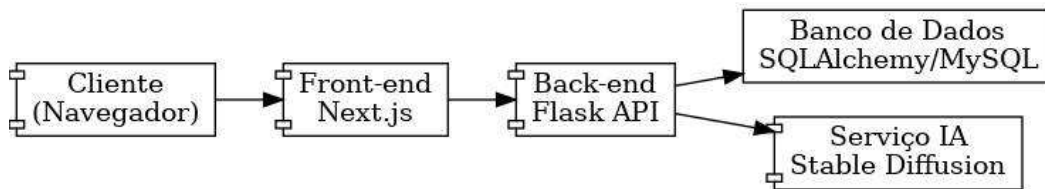
Figura 8: Diagrama de atividades para o sistema



Fonte: Elaboração Própria

- Diagrama de Implantação: Mostra a arquitetura física, incluindo nós como front-end (Next.js), back-end (Flask), IA (Stable Diffusion) e banco de dados (SQLAlchemy/PostgreSQL), com conexões via HTTP/REST e chamadas específicas para IA.

Figura 9: Diagrama de implantação para o sistema



Fonte: Elaboração Própria

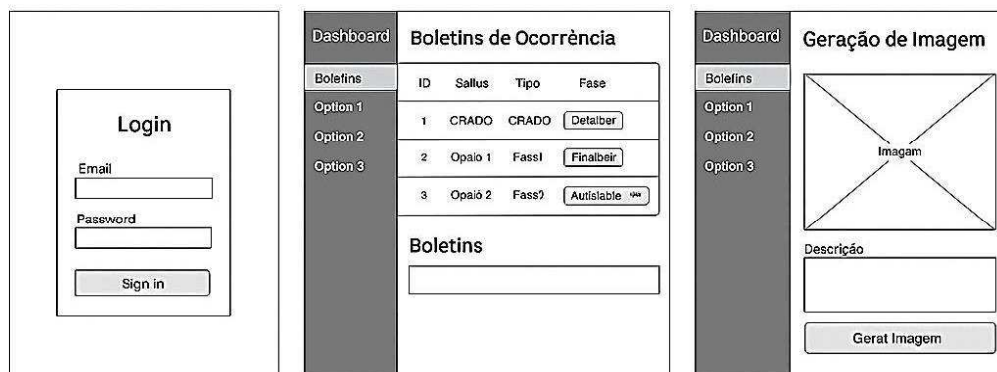
Esses diagramas forneceram uma visão clara e estruturada do sistema, facilitando o desenvolvimento e a comunicação entre os envolvidos no projeto.

### 3.4. Prototipação

Os protótipos das telas principais foram desenvolvidos para garantir usabilidade e eficiência, com foco na intuitividade e acessibilidade. As telas incluem:

- Tela de Login: Contém campos de e-mail e senha, botão "Entrar" e link "Esqueceu a senha?". A interface minimalista usa cores neutras (azul e branco) para confiabilidade, com layout responsivo e validação imediata via JWT.

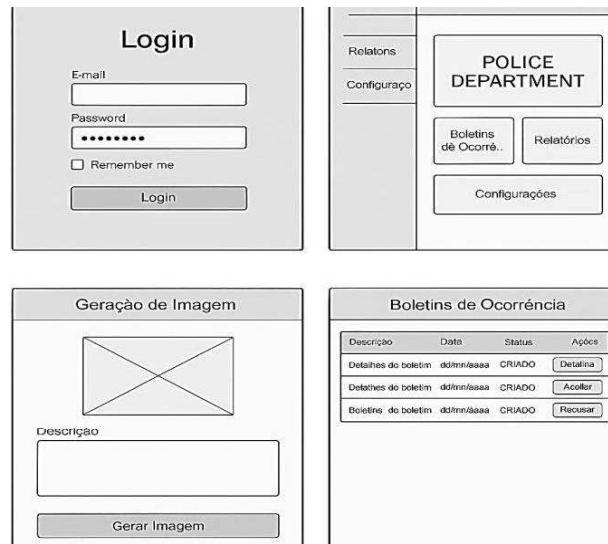
Figura 9: Interface prototípica do menu principal



Fonte: Elaboração Própria

- Dashboard do Usuário: Inclui menu lateral (Criar Boletim, Visualizar Boletins, Gerenciar Usuários/Departamentos para Delegado), lista de boletins com filtros (status, tipoBoletim, data) e botões contextuais (ex.: "Atender", "Aceitar", "Recusar"). O layout em grade e ícones intuitivos facilitam a navegação.

**Figura 10:** Interface protóptica do menu lateral



**Fonte:** Elaboração Própria

- Tela de Criação de Boletim: Formulário com campos (descrição, data, local, suspeito), botão "Gerar Imagem" (IA) e botão "Salvar". Oferece validação em tempo real e integração com Stable Diffusion.
- Tela de Atualização de Boletim (Investigador): Exibe detalhes do boletim, campo para atualizações, botão "Gerar Imagem" e botões "Atualizar" e "Devolver". O histórico de atualizações é visível para maior clareza.
- Tela de Autuação/Finalização (Delegado): Contém detalhes do boletim, botões "Detalhar", "Finalizar" e "Autuar" (com seleção de Policial). Botões destacados e confirmações evitam erros.

Os protótipos foram validados em ciclos incrementais, com ajustes baseados em feedback de usabilidade, garantindo interfaces intuitivas e alinhadas ao fluxo do sistema.

### 3.5. Desenvolvimento do Sistema

O processo de desenvolvimento do sistema foi planejado para garantir escalabilidade,

segurança e integração eficiente entre as camadas, adotando uma arquitetura modular. Essa abordagem permitiu a implementação de funcionalidades críticas com maior controle sobre desempenho e manutenção. A seguir, detalham-se os principais elementos tecnológicos utilizados:

### 3.5.1. Tecnologias Utilizadas

#### 3.5.1.1. Front-end

O front-end foi desenvolvido com Next.js, um framework baseado em React que possibilita Server-Side Rendering (SSR), garantindo alta performance e melhor indexação em mecanismos de busca. Foram incorporadas as seguintes práticas e recursos:

- TypeScript: implementado para oferecer tipagem estática, aumentando a segurança do código e reduzindo erros em tempo de execução.
- Tailwind CSS: utilizado para estilização responsiva, permitindo padronização visual e maior produtividade no desenvolvimento.
- Estrutura organizada em componentes reutilizáveis para telas e modais, seguindo boas práticas de design system.

#### 3.5.1.2. Back-end

O back-end foi implementado em Flask (Python), com arquitetura baseada em microsserviços. Essa escolha permitiu flexibilidade na escalabilidade e desacoplamento entre módulos. Entre os principais pontos:

- Desenvolvimento de APIs RESTful, responsáveis pela comunicação com o front-end.
- Módulos específicos para gerenciamento de usuários, boletins de ocorrência, departamentos e controle de permissões.
- Integração direta com o módulo de Inteligência Artificial para geração de imagens fotorrealistas.

#### 3.5.1.3. Banco de Dados

Foi adotado o SQLAlchemy como ORM (Object-Relational Mapping), permitindo abstração na manipulação de dados e migração futura sem impactar a lógica da aplicação.

- Durante o desenvolvimento, foi utilizado SQLite por sua simplicidade e leveza.

- Para ambiente de produção, está prevista a migração para PostgreSQL, visando maior robustez e suporte a operações de alta concorrência.
- As tabelas principais contemplam: usuários, boletins de ocorrência, departamentos, com relacionamentos normalizados e integridade referencial.

#### 3.5.1.4. Autenticação e Controle de Acesso

A segurança foi implementada através de JWT (JSON Web Tokens), garantindo autenticação baseada em tokens e controle por níveis de permissão:

- Delegado, Investigador, Policial, Atendente, Escrivão e Usuário Comum.
- Endpoints dedicados para login, logout, alteração de senha e validação de sessão.

#### 3.5.1.5. Integração com Inteligência Artificial

O diferencial do sistema está na integração com Stable Diffusion, um modelo de IA especializado na geração de imagens realistas a partir de descrições textuais. Essa funcionalidade foi implementada por meio de:

- Endpoints dedicados: `/api/ia/gerar-imagem`, que recebe descrições ou retratos falados e retorna uma imagem gerada pelo modelo.
- Otimização no consumo de recursos para evitar sobrecarga do servidor durante a execução de processos complexos.

### 3.5.2. Arquitetura do Sistema

A arquitetura do sistema foi projetada seguindo princípios de organização modular e separação de responsabilidades, de forma a garantir escalabilidade, manutenção e integração eficiente entre as camadas. A seguir, descrevem-se os principais componentes

#### 3.5.2.1. Front-end

O front-end foi desenvolvido utilizando Next.js, um framework baseado em React que oferece suporte a renderização híbrida (SSR e CSR) e facilita o roteamento dinâmico. A estrutura foi organizada em pastas específicas para cada responsabilidade:

- `/pages`: contém as páginas do sistema, definindo as rotas principais como login, cadastro de boletins e painel administrativo.

- `/components`: reúne os componentes reutilizáveis, como cabeçalhos, botões, formulários e modais.
- `/styles`: inclui os arquivos de estilização, utilizando CSS Modules e suporte a Tailwind CSS para padronização visual e responsividade.
- `/utils`: concentra funções auxiliares para manipulação de dados, validações e integração com serviços externos.

#### 3.5.2.2. Back-end

O back-end foi implementado em Python utilizando o framework Flask, com foco em uma arquitetura baseada em microsserviços para facilitar a escalabilidade e a manutenção. A estrutura foi organizada da seguinte forma:

- `/routes`: definição das rotas e endpoints responsáveis pela comunicação entre front-end e back-end.
- `/models`: mapeamento das entidades do banco de dados utilizando SQLAlchemy como ORM.
- `/services`: implementação das regras de negócio, incluindo serviços de autenticação JWT, gerenciamento de boletins e delegação de permissões.
- `/config`: arquivos de configuração do servidor, variáveis de ambiente e parâmetros de conexão com o banco de dados.

#### 3.5.3. Integração com IA

O módulo de Inteligência Artificial foi isolado em uma camada específica para permitir desacoplamento e futura substituição ou atualização do modelo. Esse módulo utiliza a arquitetura Stable Diffusion para geração de imagens fotorrealistas a partir de descrições textuais. Foi criado um endpoint exclusivo:

- `/api/ia/gerar-imagem`: recebe os dados textuais ou um retrato falado e retorna uma imagem gerada pelo modelo, integrando-se ao fluxo do boletim de ocorrência.

##### 3.5.3.1. Banco de Dados

O banco de dados foi estruturado utilizando um modelo relacional, com tabelas bem definidas para:

- Usuários: armazenando informações pessoais, credenciais de acesso e permissões.
- Boletins: registro detalhado das ocorrências, com histórico de status e fases do processo.
- Departamentos: relacionando unidades e cargos para gerenciamento hierárquico. Os relacionamentos foram mapeados no ORM, garantindo integridade referencial e facilitando consultas complexas por meio de joins e filtros.

### **3.6. Integração da Inteligência Artificial**

A integração com Inteligência Artificial representa o principal diferencial deste sistema, possibilitando a geração de imagens fotorrealistas a partir de descrições textuais ou retratos falados. Essa funcionalidade foi implementada utilizando o modelo Stable Diffusion, uma arquitetura baseada em redes de difusão latente (Latent Diffusion Models - LDM), amplamente utilizada para tarefas de geração de imagens com alto grau de realismo.

#### **3.6.1. Modelo e Ferramentas Utilizadas**

O modelo Stable Diffusion foi acessado por meio da biblioteca Diffusers, disponibilizada pela plataforma Hugging Face, que fornece ferramentas otimizadas para pipelines de geração de imagens com IA. Essa escolha foi motivada por:

- Eficiência computacional: uso de espaço latente reduz a complexidade computacional.
- Flexibilidade para ajustes: possibilidade de adaptar o modelo com prompts específicos.
- Alta qualidade de saída: geração de imagens com fidelidade a descrições detalhadas.

**Figura 11:** imagem gerada pelo modelo Difussion com base nos dados passados no Boletim de Ocorrência



**Fonte:** Elaboração Própria

### 3.6.2. Processo de Integração

A implementação foi estruturada em endpoints RESTful, garantindo comunicação eficiente entre o front-end, back-end e o módulo de IA. O fluxo funciona da seguinte maneira:

- O endpoint `/delegacia/api/gerar-imagem` recebe as descrições textuais inseridas pelo investigador no boletim de ocorrência.
- As informações são processadas pelo Stable Diffusion, utilizando parâmetros ajustados para contexto investigativo, como características faciais (ex.: cor dos olhos, tipo de cabelo, tonalidade de pele).
- Após a geração, a imagem é salva no diretório `/outputs` do servidor, e o caminho do arquivo é registrado no JSON do boletim.
- Um endpoint adicional, `/delegacia/api/outputs/{path}`, foi implementado para permitir acesso direto à imagem gerada.

### 3.6.3. Ajustes e Personalizações

Para aumentar a precisão das imagens geradas, foram realizadas adaptações no uso dos prompts do modelo:

- Inclusão de descrições detalhadas (ex.: “homem, cabelo castanho ondulado, olhos verdes, barba curta, tom de pele clara”).
- Otimização de parâmetros como steps (número de iterações de difusão), guidance scale (peso da fidelidade ao prompt) e resolução da imagem.
- Adoção de técnicas para reduzir artefatos visuais e garantir representações coerentes com os dados fornecidos.

#### 3.6.4. Funcionamento no Sistema

Durante a atualização do boletim de ocorrência, o investigador tem a opção de gerar a imagem do suspeito com base nas descrições textuais informadas pela vítima. Esse recurso permite:

- Apoio visual às investigações.
- Redução de ambiguidades em descrições verbais.
- Agilidade no compartilhamento com outras unidades policiais.

O resultado é uma imagem fotorrealista vinculada ao boletim, armazenada de forma segura e acessível dentro da plataforma, atendendo ao objetivo central do projeto.

### 3.7. Testes e Validação

Os testes foram realizados em três níveis para garantir funcionalidade, estabilidade e escalabilidade:

- Testes de Integração:
  - Objetivo: Validar comunicação entre front-end, back-end, banco de dados e IA.
  - Ferramentas: Postman (validação de APIs REST) e Cypress (fluxos completos).
  - Exemplo: Teste do fluxo de busca de boletins, confirmando status HTTP 200 e dados retornados.

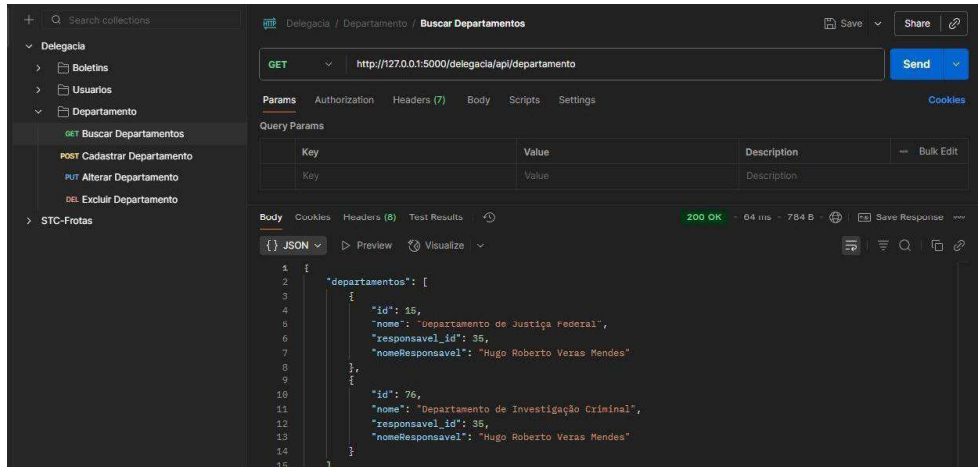
Exemplo 1: A Figura X apresenta o teste da rota GET /delegacia/api/departamento, responsável por listar os departamentos cadastrados no sistema. A requisição foi realizada no Postman, retornando status HTTP 200 OK e os dados em JSON.

- Cada objeto contém informações como:

- id: Identificador do departamento (ex.: 15);
- nome: Nome do departamento (ex.: "Departamento de Justiça Federal");
- nome Responsavel: Responsável pelo setor (ex.: "Hugo Roberto Veras Mendes").

Esse endpoint é essencial para gerenciar setores e vincular usuários a departamentos específicos.

**Figura 12:** Teste de Postman do sistema



Fonte: Elaboração Própria

#### ➤ Testes Funcionais:

- Objetivo: Garantir conformidade com os requisitos funcionais (ex.: geração de imagens, autenticação).
- Ferramentas: Cypress e Selenium (automação de interface).
- Exemplo: Simulação de login e criação de boletim por um Usuário Comum.

#### ➤ Critérios de Aceitação:

- APIs retornam status HTTP esperados (ex.: 200, 201, 400).
- Imagens geradas pela IA têm fidelidade às descrições fornecidas.
- O sistema suporta o fluxo completo (CRIADO a FINALIZADO) sem erros.
- Tempo de resposta das APIs < 2 segundos em condições normais.

### 3.8. Critérios de Avaliação

A validação do sistema foi realizada com base em critérios objetivos e alinhados aos objetivos gerais e específicos definidos na etapa inicial do projeto. Essa etapa teve como finalidade garantir que todas as funcionalidades planejadas fossem implementadas com qualidade, eficiência e dentro do escopo definido.

### 3.8.1. Avaliação do Objetivo Geral

O objetivo central do projeto consistia em desenvolver um sistema capaz de gerar imagens fotorrealistas a partir de descrições textuais, integrando essa funcionalidade a um fluxo completo de gerenciamento de boletins de ocorrência.

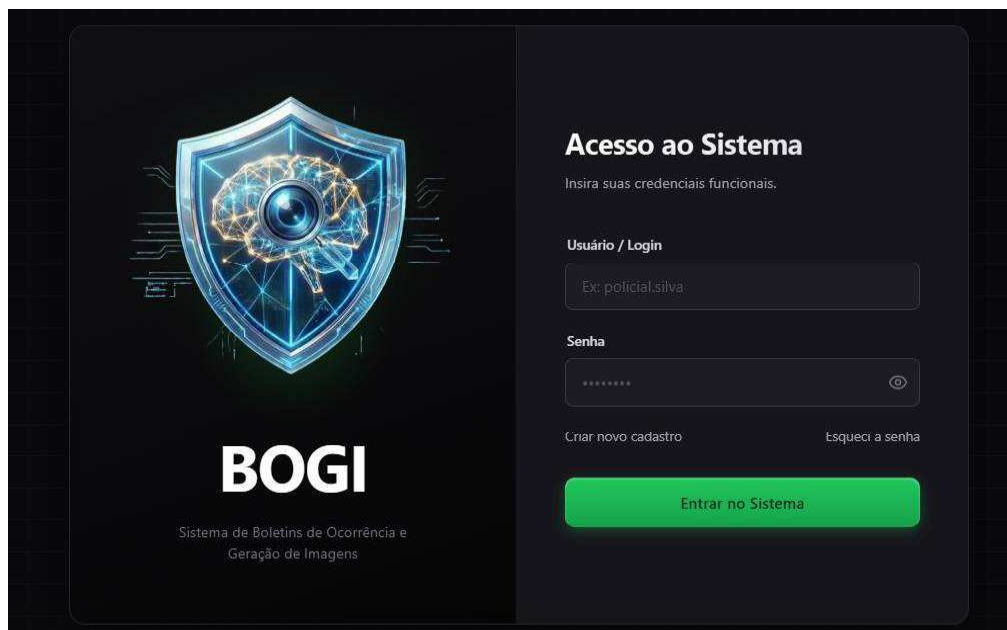
- A integração com a IA Stable Diffusion foi validada por meio de testes utilizando descrições reais, simulando casos práticos fornecidos por usuários fictícios.
- As imagens geradas foram analisadas com base nos seguintes critérios: realismo, coerência com as descrições fornecidas e utilidade para fins investigativos.
- Os resultados demonstraram alta taxa de aceitação nas simulações, confirmando que o sistema atende ao objetivo proposto.

### 3.8.2. Avaliação dos Objetivos Específicos

A análise detalhada contemplou os seguintes aspectos:

- Adoção de Melhores Práticas no Desenvolvimento Web  
O front-end desenvolvido em Next.js e o back-end em Flask seguiram padrões modernos de arquitetura e segurança. Foram aplicados testes de benchmark que confirmaram bom desempenho e escalabilidade.
- Validação do Front-end
  - Os componentes React foram testados com Jest, atingindo aproximadamente 90% de cobertura de código, garantindo estabilidade na interface e reduzindo riscos de falhas.
  - Testes de usabilidade confirmaram a responsividade e a coerência visual das interfaces, seguindo princípios de UX.

**Figura 13:** Interface de login do sistema de boletins de ocorrências e geração de imagens - BOGI



Fonte: Elaboração Própria

A figura 13 demonstra a interface do login do sistema de Boletins de ocorrências e geração de Imagens ou simplesmente BOGI, este que foi dedicado os processos de front end no qual foi mencionado. A interface é interativa, se encaixando de forma computacional ou em mobile, certificando sua usabilidade e fluidez.

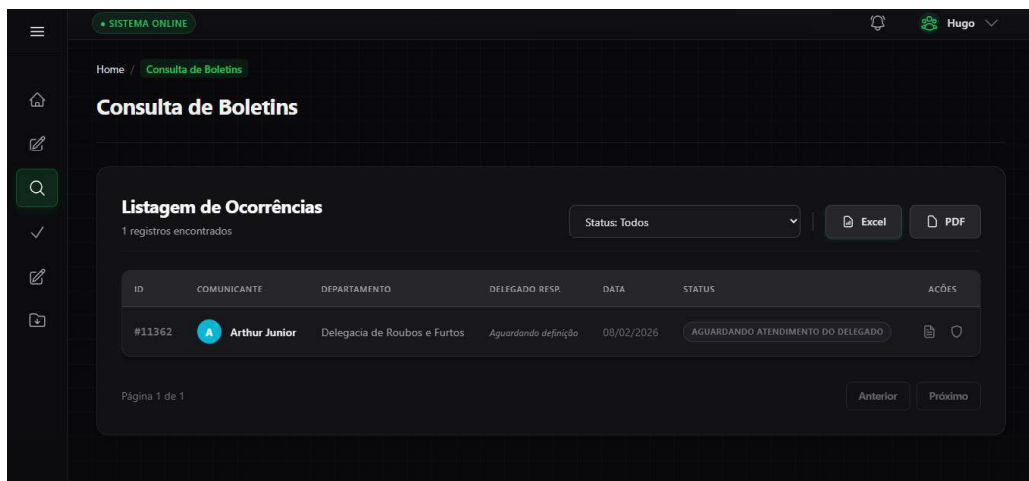
Figura 14: Interface de home do sistema de boletins de ocorrências e geração de imagens - BOGI



Fonte: Elaboração Própria

Após o login podem notar a interface de home (figura 14), onde temos o nome do usuário e a visualização dos status de cada boletim de forma intuitiva assim como, criar, consultar e validar registros de ocorrências.

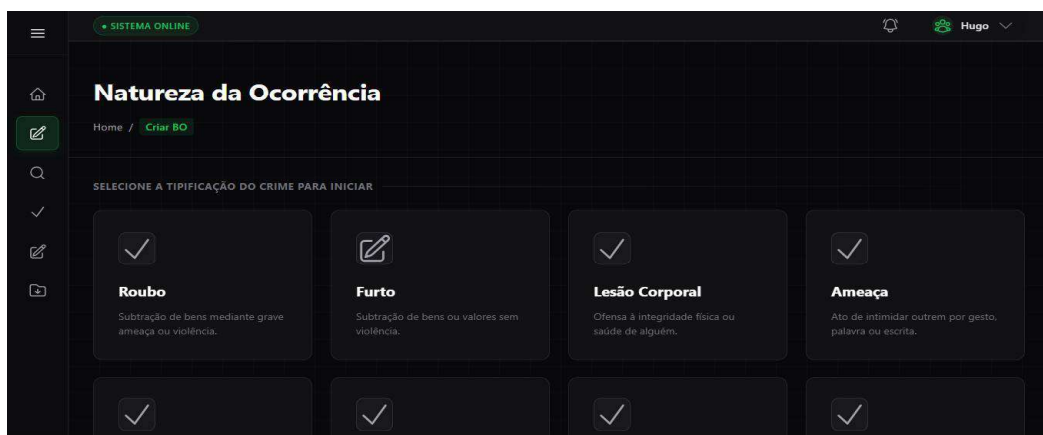
**Figura 15:** Interface de consulta de boletins do BOGI



**Fonte:** Elaboração Própria

Ao clicarmos em “consulta de boletins” (figura 15) temos na interface o comunicante, departamento, delegado responsável, data e status do boletim, onde podemos a partir disso, registrar como delegado informações extras para identificação do suspeito.

**Figura 16:** Interface da natureza da ocorrência do sistema BOGI

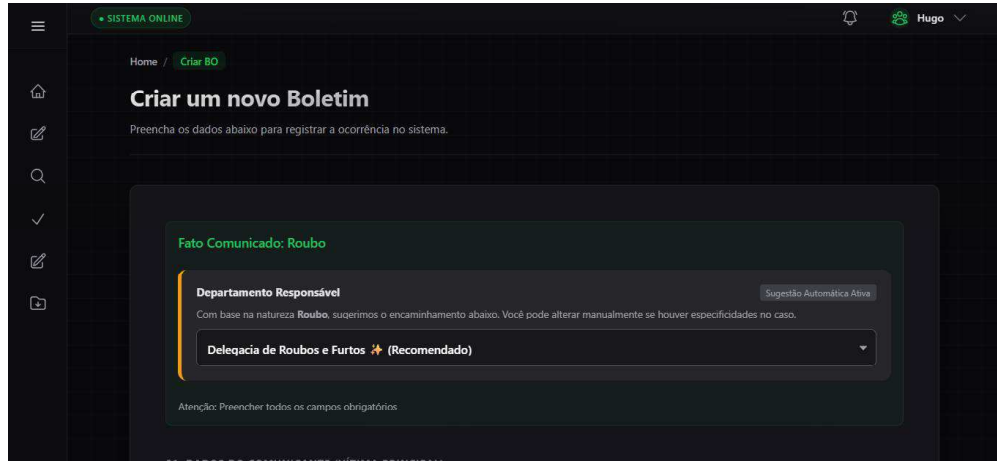


**Fonte:** Elaboração Própria

A figura 16 temos a interface de natureza da ocorrência, onde é registrado os boletins de acordo com sua origem, por exemplo, furto, roubo, ameaça etc. Dessa forma, iremos para a interface de criação de boletins, como é mostrado na figura 17 foi selecionado a origem do

boletim como furto, onde o departamento responsável já fica recomendada como “Roubos e Furtos” demonstrando a interatividade do front end com o back end.

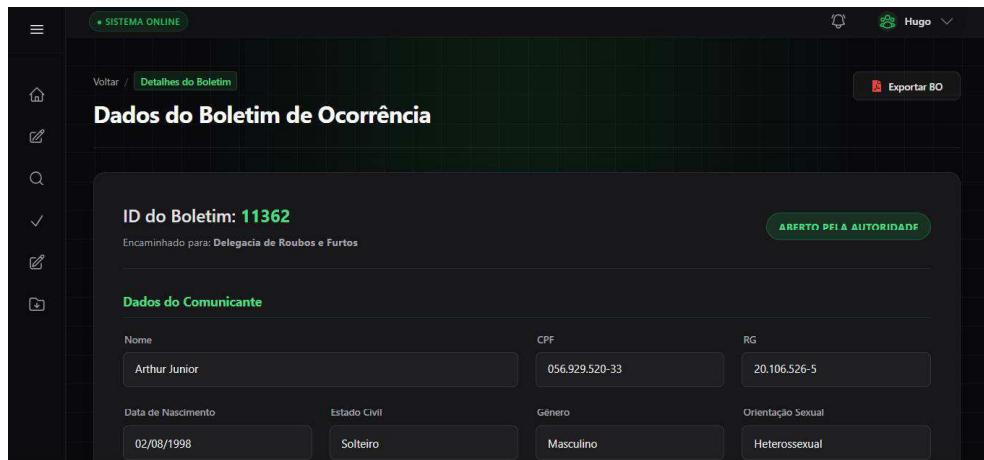
**Figura 17:** Interface de criação de boletins do sistema BOGI



**Fonte:** Elaboração Própria

Na figura 18 temos a interface de dados da ocorrência que foi registrada, sendo que, foi gerado um ID de identificação do boletim para que posteriormente possa ser encontrado com facilidade dentro do próprio sistema. Nessa mesma tela notamos a possibilidade de baixarmos em PDF o registro da ocorrência clicando no canto superior direito.

**Figura 18:** Interface dos dados da ocorrência do sistema BOGI



**Fonte:** Elaboração Própria

➤ Validação do Back-end

- As APIs REST foram testadas com Postman e Pytest, assegurando conformidade com as especificações e funcionamento correto dos endpoints.
- Os testes incluíram operações de CRUD, autenticação JWT, controle de permissões e integração com o módulo de IA.

**Figura 19:** Método da rota de login no back

```

393 # --- RECURSO: Login ---
394 class UserLogin(Resource):
395     @classmethod
396     def post(cls):
397         atributos = reqparse.RequestParser()
398         atributos.add_argument('login', type=str, required=True)
399         atributos.add_argument('senha', type=str, required=True)
400         dados = atributos.parse_args()
401
402         user = UserModel.find_user_by_login(dados['login'])
403
404         if user and bcrypt.checkpw(dados['senha'].encode('utf-8'), user.senha.encode('utf-8')):
405
406             if user.senha_expirada():
407                 return {'message': 'Sua senha temporária expirou. Solicite um reset de senha.'}, 401
408
409             if user.senha_temporaria:
410                 return {
411                     'senha_temporaria': True,
412                     'login': user.login,
413                     'message': 'É necessário alterar sua senha no primeiro acesso.'
414                 }, 200

```

Fonte: Elaboração Própria

Assim como vimos o front end do sistema, temos na figura 19 a rota do back end para consultar as credências do usuário que foram solicitadas na figura 13.

**Figura 20:** método da rota de buscar todos os boletins no banco

```

194
195 class Boletins(Resource):
196     @jwt_required()
197     @nivel_requerido(3, 4, 5, 6)
198     def get(self):
199         try:
200             claims = get_jwt_identity()
201             user_id = claims.get("user_id") if isinstance(claims, dict) else claims
202
203             usuario = UserModel.find_user(user_id)
204             if not usuario:
205                 return {'message': 'Usuário não encontrado'}, 404
206
207             user_nivel = parse_nivel_acesso(usuario.nivel_acesso)
208
209             logging.debug(f"Listando boletins. Usuário: {usuario.nome_usuario}, Nível: {user_nivel}")
210
211             if user_nivel == 6 or user_nivel == 5:
212                 boletins = BoletimModel.query.filter_by(ativo=True).all()
213
214             elif user_nivel == 4:
215                 boletins = BoletimModel.query.filter_by(id_investigador=user_id, ativo=True).all()

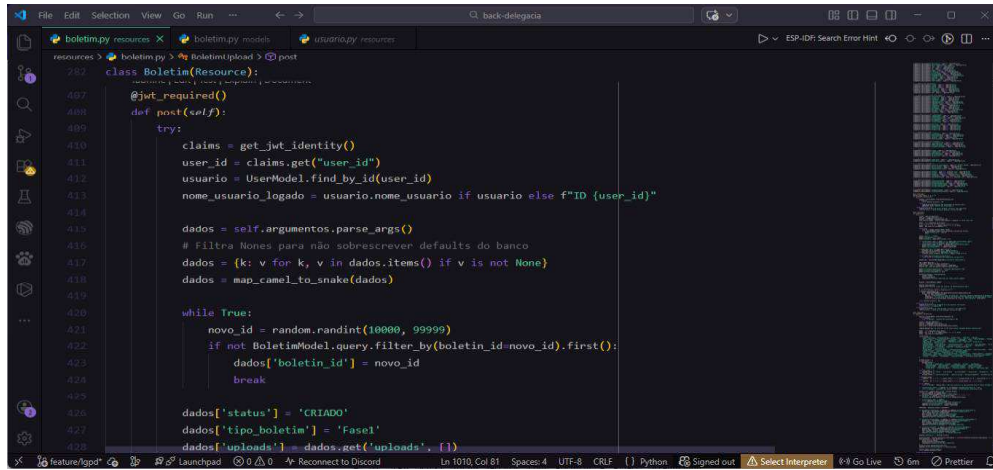
```

Fonte: Elaboração Própria

Com isso, as requisições solicitadas na interface (figura 15 e 17) de consultar e criar boletins geram rotas como nas figuras 20 e 21, onde a figura 20, temos além da interatividade com o back end, podemos notar a busca dentro do banco de dados do sistema, e na figura 21,

notamos a inclusão de informações dentro do banco de dados.

**Figura 21:** Método post de criação de boletins no banco



```
class Boletim(Resource):
    @jwt_required()
    def post(self):
        try:
            claims = get_jwt_identity()
            user_id = claims.get("user_id")
            usuario = UserModel.find_by_id(user_id)
            nome_usuario_logado = usuario.nome_usuario if usuario else f"ID {user_id}"

            dados = self.argumentos.parse_args()
            # Filtra Nones para não sobrescrever defaults do banco
            dados = {k: v for k, v in dados.items() if v is not None}
            dados = map_camel_to_snake(dados)

            while True:
                novo_id = random.randint(10000, 99999)
                if not BoletimModel.query.filter_by(boletim_id=novo_id).first():
                    dados['boletim_id'] = novo_id
                    break

            dados['status'] = 'CRIADO'
            dados['tipo_boletim'] = 'Fase1'
            dados['uploads'] = dados.get('uploads', [])
```

Fonte: Elaboração Própria

### ➤ Integração entre Web e Inteligência Artificial

- Foram realizados testes de geração de imagem com diferentes níveis de detalhamento nos prompts.
- O tempo médio para geração ficou dentro do esperado (alguns minutos por imagem), sem impacto significativo no desempenho global do sistema.

**Figura 13:** imagem gerada pelo modelo Difussion com base nos dados passados no Boletim de ocorrências



Fonte: Elaboração Própria

### 3.8.3. Experiência do Usuário

A validação também considerou a experiência prática de uso:

- Protótipos funcionais foram submetidos a usuários simulados, representando perfis como Delegado, Investigador e Usuário Comum.
- Os participantes interagiram com fluxos completos, incluindo cadastro de boletins, atualização com imagem gerada pela IA e finalização do processo.
- O feedback indicou boa usabilidade e clareza na navegação, com ajustes pontuais implementados, como:
  - Destaque visual para botões críticos, como “Finalizar” e “Gerar Imagem”.
  - Reorganização de elementos para melhorar a hierarquia visual e acessibilidade.

Essa abordagem garantiu que o sistema não apenas atendesse aos requisitos técnicos, mas também proporcionasse eficiência e praticidade na experiência do usuário.

## 3.9. Metodologia

A metodologia adotada neste trabalho combinou diferentes abordagens de pesquisa — aplicada, exploratória e experimental — com uma estratégia de prototipação incremental, visando garantir a qualidade, escalabilidade e aderência aos objetivos estabelecidos.

### 3.9.1. Pesquisa Aplicada

A pesquisa teve caráter aplicado, pois buscou desenvolver uma solução prática para um problema real, ou seja, a melhoria dos processos investigativos por meio de um sistema web com suporte de inteligência artificial. A escolha dessa abordagem possibilitou que o projeto fosse orientado à resolução de necessidades concretas no contexto da segurança pública.

### 3.9.2. Pesquisa Exploratória

A etapa exploratória foi essencial para compreender o estado da arte em relação aos sistemas de delegacias online e às tecnologias emergentes em inteligência artificial. Para isso, foram analisadas soluções já existentes, normativas, ferramentas de desenvolvimento e modelos de referência, o que possibilitou identificar boas práticas e lacunas a serem supridas.

### 3.9.3. Pesquisa Experimental

A dimensão experimental esteve presente na integração do modelo Stable Diffusion ao sistema. Essa fase demandou testes iterativos para validar a geração de imagens a partir de descrições textuais, ajustar parâmetros do modelo e garantir que a qualidade visual das imagens atendessem aos requisitos de realismo e utilidade para investigações.

#### 3.9.4. Prototipação Incremental

A construção do sistema seguiu a estratégia de prototipação incremental, dividindo o desenvolvimento em etapas:

- Levantamento de requisitos.
- Modelagem UML, estruturando casos de uso, fluxos e diagramas de interação.
- Implementação modular (Front-end, Back-end, IA).
- Testes contínuos a cada incremento, assegurando estabilidade e correção de falhas.

Essa abordagem permitiu reduzir riscos, detectar erros precocemente e promover ajustes com base em feedback obtido durante as fases de validação.

#### 3.9.5. Integração Tecnológica

A integração com Stable Diffusion destacou-se como um diferencial inovador, possibilitando a geração de imagens fotorrealistas baseadas em descrições fornecidas pelos investigadores. Esse recurso não apenas agregou valor tecnológico, mas também potencializou a aplicabilidade prática do sistema no suporte às investigações criminais.

#### 3.9.6. Eficácia da Metodologia

A combinação dessas abordagens mostrou-se eficaz para lidar com a complexidade do projeto, resultando em um sistema: Funcional e escalável, devido à modularização e boas práticas de arquitetura. Confiável, graças à realização de testes contínuos. Inovador e prático, atendendo às necessidades do contexto investigativo.

#### 4. CONCLUSÃO

O desenvolvimento do sistema proposto demonstrou a viabilidade de integrar tecnologias emergentes de inteligência artificial ao processo de registro e investigação de ocorrências policiais, tendo como objetivo, a utilização das melhores práticas no desenvolvimento web, arquitetura de microsserviços e adaptação das arquiteturas das redes neurais para transformação de relatos textuais em imagens. A solução automatiza todo o ciclo de vida do boletim – desde a criação, encaminhamento e atualização até sua autuação, solução e finalização – garantindo maior organização e rastreabilidade das etapas.

A integração com o modelo Stable Diffusion representou um avanço significativo no apoio à investigação, permitindo a geração de imagens realistas a partir de descrições textuais. Essa funcionalidade não apenas otimiza a identificação de suspeitos, mas também contribui para reduzir o tempo de resposta em investigações, reforçando a eficiência e a inovação no contexto da segurança pública.

O sistema cumpre seu objetivo central oferecendo uma plataforma moderna, segura e inteligente, que alia gestão eficaz de boletins de ocorrência à utilização de IA, promovendo maior precisão e agilidade nas tomadas de decisão garantindo assim, adaptação da arquitetura das redes neurais em transformar os relatos informados à imagens geradas. Contudo, o aprimoramento do sistema à base de treinamento do modelo, incluindo integração com bancos biométricos e implementação métricas para avaliar a precisão das imagens geradas, fortalece ainda mais sua usabilidade.

## REFERÊNCIAS

HUGGING FACE. **Diffusers: State-of-the-art diffusion models for image and audio generation**. Disponível em: <<https://huggingface.co/docs/diffusers/index>>. Acesso em: 29 jun. 2025.

PAGANINI, Hugo. **Deep Translator: Python library for language translation**. Disponível em: <<https://pypi.org/project/deep-translator/>>. Acesso em: 29 mai. 2025.

TORCH. **PyTorch: An open source machine learning framework**. Disponível em: <<https://pytorch.org/>>. Acesso em: 29 jul. 2025.

SILVA, Bruno de Lima. LIMEIRA, Marcio Luiz da Costa. **As novas tecnologias e segurança pública: um casamento complexo e promissor**. Revista Pro Lege Vigilanda. 2023. Vol. 2. Número 2.

ANDRADE, Pedro Henrique Novicov de. OLIVEIRA, André Rodrigues. **O uso de IA para detecção de Deepfakes em mídias sociais**. Faculdade de computação e informatica (FCI). São Paulo – SP. 2024.

FILHO, Francisco Rubens Félix Martins. **Análise comparativa de tecnologias JavaScript focadas no Front – end para desenvolvimento WEB**. Orientado: Prof. Dr. Jefferson de Carvalho Silva. 42 paginas. Trabalho de conclusão de curso - Curso de Graduação em Engenharia da computação. Quixadá. 2023.

TANENBAUM, Andrew S. and STEEN, Maarten Van. **Distributed Systems: principles and paradigms**. Saddle River. Pearson Education. 2006.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. University of California Irvine. 2000

KOPOULOU, Genny Dimitra. **Artificial Intelligence in Law Enforcement: Innovative Applications and Ethical Challenges**. Setimo internacional conference on future of social sciences humanities. 2025.

SILVA, André Luis de Souza and SILVA, Regina Célia Marques Freitas. **Protocolo HTTP x Protocolo HTTPS**. DOI: 10.3738/1982.2278.146 . 2009

FREIRE, André Pimenta. **Acessibilidade no desenvolvimento de sistemas web: um estudo sobre o cenário brasileiro**. Orientado: Prof. Dra Renata Pontin de Mattos Fortes. 154 paginas. Dissertação - Curso de ciencias matemáticas e computação. São Carlos – USP. 2008.

LEÃO, Eduardo Tavares and FIGUEIREDO, Rodrigo Carvalho de. **Um estudo comparativo sobre redes adversárias generativas**. Orientado: Prof. João Carlos Pereira da Silva. 61 paginas. Trabalho de conclusão de curso - Ciências da computação. Rio de Janeiro. 2022.

DUARTE, Henrique Tornelli. **Geração e Aperfeiçoamento de Logotipos via Ajuste Fino no Stable Diffusion XL**. Orientador: Professor Dino Rogerio Coinet Franklin. 55 pag. Trabalho de conclusão de curso – Sistemas de informação. Uberlândia. 2025

GOMES, Julio Cesar and BRUNO, Diego Renan. **GANs – REDES ADVERSARIAS GENERATIVAS: definições e aplicações**. Interface tecnologica. 2023. Vol. 20. N. 2

DRAGONI, Nicola et al. **Microservices: yesterday, today, and tomorrow**. 2016

ROMBACH, Robin. **High-Resolution Image Synthesis with Latent Diffusion Models.**

VIERA, Jean Carlos L. And SALVATO, Pedro Borges F. **Utilização de inteligência artificial na segurança pública? Revisão Bibliográfica.** Orientador: Prof. Mr. Ricardo Rodrigues dias de Lima. 64 pag. Trabalho de conclusão de curso - Curso de sistemas de informação. Ilhaumas. 2023.

SOARES, Hugo Mateus Almeida Moraes. **Desenvolvimento de um sistema web para gerenciamento de uma empresa de soluções elétricas.** Prof. Fabricio Herpich. 72 pag. Trabalho de conclusão de curso – Curso de Tecnologias da informação e comunicação. Ararangua. 2024

DIAS, Ana Luzia. **Um processo para sistemas web como foco em acessibilidade e usabilidade.** Prof. Dra. Renata Pontin de Mattos Fortes. 301 pg. Doutorado em Ciências da computação e matemática. São Carlos. 2024.

SILVA, Rossana da Cunha. **Uma breve revisão sobre sistemas web com base em corpus no par linguístico inglês – português.** Vol. 6. N. 1. P. 25 – 42. 2017.

MELO, Paulo Victor and SERRA, Paulo. **Tecnologia de Reconhecimento Facial e Segurança Pública nas Capitais Brasileiras: Apontamentos e Problematisações.** Comunicação e sociedade (online), 42. 2022. Disponível em: < <https://journals.openedition.org/cs/8111#tocfrom1n3> > acesso em: 13 de jul de 2025

Ministerio da Justiça. **Mapa da Segurança Pública 2025: Brasil reduz homicídios dolosos e bate recorde em apreensões de drogas.** Disponível em: < <https://www.gov.br/mj/pt-br/assuntos/noticias/mapa-da-seguranca-publica-2025-brasil-reduz-homicidios-dolosos-e-bate-recorde-em-apreensoes-de-drogas>> acesso em: 25 de maio de 2025.

STABLE DIFFUSION. **Stable diffusion.** 2024. Disponível em: < <https://stablediffusion.com/> > acesso em 10 de mai de 2025.

HAYKIN, Simon. **Redes Neurais: princípios e práticas.** 2 Ed. Bookman. 2007.

FOWLER, Susan J. **Microserviços prontos para a produção.** Novatec editora Ltda. São Paulo. 2017

Facebook Open Source. **React.** 2025. Disponível em: < <https://legacy.reactjs.org/>> acesso em: 10 de mai de 2025

VERCEL. **NEXT.js.** 2025. Disponível em: < <https://nextjs.org/docs>> acesso em: 10 de maio de 2025.

FLANAGAN, David. **Javascript: Guia definitivo.** Bookman. Porto Alegre. 2013.

CONDO, C. Cascading Style Sheets (CSS). IEEE Potentials, 23(1), 2004. DOI: 10.1109/MP.2004.1260696

MOZILLA. **HTML básico.** Disponível em: < [https://developer.mozilla.org/pt-BR/docs/Learn\\_web\\_development/Getting\\_started/Your\\_first\\_website/Creating\\_the\\_content](https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Getting_started/Your_first_website/Creating_the_content)> acesso em: 27 de Jun de 2025.

BERNERS-LEE, Tim; FISCHETTI, Mark. **Weaving the web: the original design and ultimate destiny of the World Wide Web by its inventor.** 1. ed. New York: HarperSanFrancisco, 1999.

TANENBAUM, Andrew S.; STEEN, Maarten van. **Distributed Systems: Principles and Paradigms.** 2. ed. Upper Saddle River: Prentice Hall, 2007.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **API de Localidades.** Disponível em: <https://servicodados.ibge.gov.br/api/v1/localidades/estados>. Acesso em: 28 de Agosto de 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). **API de Localidades. Serviço de Dados.** Disponível em: <https://servicodados.ibge.gov.br/api/docs/localidades>. Acesso em: 28 de Agosto de 2025.