



engenheiro de software
técnico em eletroeletrônica

NATAN OGLIARI

**PROGRESSO DE IMPLANTAÇÃO DA PLATAFORMA INTEGRADA DE
AÇÕES PARA ENFRENTAMENTO À VIOLÊNCIA DOMÉSTICA E
CRIMES SEXUAIS**

Maravilha/SC
31 de janeiro de 2026

NATAN OGLIARI

**PROGRESSO DE IMPLANTAÇÃO DA PLATAFORMA INTEGRADA DE
AÇÕES PARA ENFRENTAMENTO À VIOLÊNCIA DOMÉSTICA E
CRIMES SEXUAIS**

Documento elaborado para prestar esclarecimentos sobre o andamento da implantação do Progresso de implantação da **Plataforma Integrada de Ações para Enfrentamento à Violência Doméstica e Crimes Sexuais** .

.

Maravilha/SC

31 de janeiro de 2026

Sumário

	Páginas
1 Introdução	6
2 Violência Doméstica no Brasil: Panorama e Dados	6
3 Tecnologias utilizadas	8
3.1 Framework: Django	8
3.2 Banco de dados: PostgreSQL	8
3.3 Nginx	9
3.4 Tailwind CSS	9
3.5 HTMX	9
3.6 Django Channels e Redis	9
3.7 Smart Selects	10
3.8 Chart.js	10
3.9 Leaflet.js com OpenStreetMap	10
3.10 jQuery	10
3.11 Ollama: Execução Local de Modelos de Linguagem	11
4 Níveis de Prontidão Tecnológica (TRL)	11
4.1 Classificação dos Níveis TRL	11
4.2 Classificação da escala de nível de prontidão tecnológica (TRL)	12
5 Implementação para testes e validações	14
5.1 Dos Scripts	14
5.2 Hostinger	15
5.2.1 Aplicação SaaS	16
5.2.2 O Banco de dados	16
5.2.3 O domínio	16
5.3 Do Desenvolvimento	17
6 Métodos	19
6.1 Estrutura de Aplicações Django	19
6.2 Visualizações e funcionalidades	20
6.2.1 Página Inicial	20
6.2.2 Polícia Penal	20
6.2.3 Polícia Militar	21
6.3 Técnicas e Práticas de Desenvolvimento	22

7	Resultados	25
7.1	Métricas de Performance e Escalabilidade	26
7.1.1	Otimizações de Performance	26
7.2	Volume de Dados em Ambiente de Testes	27
7.3	Assistente virtual com IA	32
8	Sugestões/Para implementações futuras	34
9	Evolução do Desenvolvimento	35
9.1	Ciclo de Desenvolvimento	35
10	Considerações Finais	36
11	Anexos	40

Lista de Algoritmos

1	Interação com o Modelo llama3:70b, tempo de resposta: 224.82 [s]	34
---	--	----

Lista de Figuras

1	logo PIEVDCS.	6
2	Consulta da Polícia Militar com paginação e sem otimização	28
3	Telas da plataforma	29
4	Telas estatísticas	30
5	Telas institucionais	31
6	Modelo llama3:70b	33
7	Diploma do Bacharelado	40
8	Diploma de Técnico	41

1 Introdução

Figura 1. *logo PIEVDCS.*

A Plataforma Integrada de Informações para Enfrentamento à Violência Doméstica e Crimes Sexuais (PIEVDCS) é uma solução digital desenvolvida com base no *framework* Django, no modelo SaaS (Software como Serviço). Seu objetivo é integrar instituições como segurança pública, Ministério Público, Poder Judiciário, Defensoria Pública e serviços municipais de assistência, promovendo agilidade e efetividade no atendimento às vítimas e concientização dos agressores.



Fonte: O autor (2025)

Além de promover agilidade e efetividade no atendimento às vítimas, a plataforma representa uma ferramenta estratégica para romper o ciclo intergeracional da violência, que frequentemente se perpetua de pais para filhos. Estudos como o da Universidade Federal do Ceará, em parceria com a ONU Mulheres, revelam que quatro em cada dez mulheres que cresceram em lares violentos vivenciam o mesmo padrão na vida adulta, evidenciando a urgência de intervenções sistêmicas e integradas (CARVALHO; OLIVEIRA, 2017). Ao conectar instituições e facilitar o fluxo de informações, a plataforma não apenas melhora o presente - ela transforma o futuro, criando condições para que novas gerações cresçam em ambientes mais seguros, protegidos e livres da normalização da violência doméstica.

O desenvolvimento da plataforma foi viabilizado por meio de verba oriunda do Poder Judiciário da Comarca de Maravilha/SC. Um aspecto singular do projeto é a participação direta de um reeducando do Presídio Regional de Maravilha como responsável técnico pelo desenvolvimento da solução. O reeducando, com formação técnica, atua sob supervisão institucional, aplicando seus conhecimentos de forma produtiva e qualificada, demonstrando que a ressocialização é viável e impactante.

Este relatório documenta o progresso técnico da implementação, destacando as funcionalidades já desenvolvidas, os métodos utilizados e as etapas pendentes até a entrega final.

2 Violência Doméstica no Brasil: Panorama e Dados

A violência doméstica contra a mulher constitui um grave problema de saúde pública e violação dos direitos humanos no Brasil. Segundo dados do Fórum Brasileiro de Segurança Pública (FBSP), em 2023 foram registradas 1.467 mortes violentas intencionais de mulheres,

sendo 722 feminicídios (Fórum Brasileiro de Segurança Pública, 2024). A cada hora, aproximadamente 46 mulheres sofrem algum tipo de agressão no Brasil, totalizando mais de 400 mil casos de violência doméstica registrados anualmente.

A Lei nº 11.340/2006, conhecida como Lei Maria da Penha, representa um marco legal fundamental no enfrentamento à violência doméstica e familiar contra a mulher. A legislação reconhece cinco formas de violência: física, psicológica, sexual, patrimonial e moral, estabelecendo mecanismos de proteção e assistência às vítimas (BRASIL, 2006). Desde sua promulgação, observou-se um aumento significativo nas denúncias, reflexo tanto da maior conscientização social quanto da ampliação dos canais de atendimento.

Dados do Instituto Maria da Penha revelam que 89% das vítimas de violência doméstica no Brasil são mulheres, sendo que 43% dos casos ocorrem dentro de casa. O parceiro íntimo (companheiro, marido, namorado ou ex) é o principal agressor em 78% dos casos registrados (Instituto Maria da Penha, 2023). Além disso, 68% das mulheres em situação de violência convivem com o agressor, evidenciando a complexidade das relações de dependência emocional, econômica e familiar.

O Anuário Brasileiro de Segurança Pública de 2024 aponta que as medidas protetivas de urgência (MPU) continuam sendo uma das principais ferramentas de proteção às vítimas. Em 2023, foram concedidas mais de 362 mil medidas protetivas no país, representando um aumento de 17% em relação ao ano anterior (Fórum Brasileiro de Segurança Pública, 2024). No entanto, desafios persistem na efetividade do monitoramento e na articulação entre as instituições responsáveis pela proteção das vítimas.

A pandemia de COVID-19 agravou o cenário da violência doméstica no Brasil. Durante o período de isolamento social em 2020 e 2021, houve um aumento de 28% nas denúncias de violência contra mulheres, segundo o Ligue 180 (Central de Atendimento à Mulher). O confinamento forçado intensificou a convivência com agressores e dificultou o acesso das vítimas às redes de proteção e denúncia (Fórum Brasileiro de Segurança Pública, 2021).

No contexto de Santa Catarina, dados da Secretaria de Estado da Segurança Pública mostram que em 2023 foram registrados 38.456 casos de violência doméstica, sendo 21.234 casos de lesão corporal dolosa e 15.879 casos de ameaça (Secretaria de Estado da Segurança Pública de Santa Catarina, 2024). O estado tem implementado políticas públicas integradas, incluindo casas-abrigo, centros de referência e patrulhas especializadas no atendimento à mulher.

A violência psicológica, frequentemente subnotificada, afeta profundamente a saúde mental das vítimas. Estudos indicam que 85% das mulheres em situação de violência doméstica desenvolvem transtornos psicológicos, como depressão, ansiedade e síndrome do pânico (ZAN-CAN; WASSERMANN; LIMA, 2013). A violência também impacta diretamente as crianças expostas ao ambiente violento, perpetuando o ciclo intergeracional já mencionado.

Diante desse cenário alarmante, a implementação de sistemas integrados de informação, como a PIEVDCS, torna-se essencial para aprimorar a coordenação entre as instituições, acelerar o atendimento às vítimas, monitorar a efetividade das medidas protetivas e produzir dados

estratégicos para o planejamento de políticas públicas de prevenção e enfrentamento à violência doméstica.

3 Tecnologias utilizadas

A escolha das tecnologias adotadas neste projeto foi pautada, sobretudo, por seu caráter *Open Source* (código aberto), o que proporciona significativa economia ao eliminar custos com licenciamento. Dessa forma, os investimentos se concentram exclusivamente em aspectos de infraestrutura, como servidores e equipe técnica especializada.

Além disso, o desenvolvedor considera que as informações sensíveis contempladas por esta plataforma devem permanecer sob responsabilidade do Estado, não devendo ser compartilhadas com empresas privadas, como serviços de hospedagem em nuvem ou soluções de inteligência artificial de terceiros. Ressalta-se, contudo, que, na fase de validação, a plataforma está hospedada em serviços de terceiros.

3.1 Framework: Django

Segundo Django (2025), Django é um framework de desenvolvimento web escrito em Python, amplamente reconhecido por acelerar a criação de aplicações robustas e escaláveis. Seu diferencial está na abordagem *"batteries included"*, oferecendo nativamente funcionalidades essenciais como autenticação, painel administrativo, mapeamento objeto-relacional (ORM) e roteamento de URLs (*Uniform Resource Locator*).

Outro fator decisivo na escolha do Django é sua natureza nativa em Python, o que possibilita fácil integração com bibliotecas de *machine learning*¹, como `scikit-learn`, `TensorFlow`, `PyTorch`, `Ollama`, dentre outras. Isso viabiliza, em futuras versões do sistema, a implementação de modelos preditivos capazes de analisar parâmetros operacionais e sugerir ações estratégicas de maneira automatizada e inteligente.

3.2 Banco de dados: PostgreSQL

PostgreSQL (2025) é um sistema de gerenciamento de banco de dados objeto-relacional de código aberto, com mais de 35 anos de desenvolvimento ativo, considerado um dos mais avançados e robustos do mercado. Conta com integração nativa ao Django.

Em termos de gerenciamento de dados, o PostgreSQL oferece suporte a extensões como PostGIS, que adiciona funcionalidades geoespaciais essenciais para a análise e visualização de dados geográficos, como mapas de ocorrências de violência doméstica. Além disso, sua arquitetura robusta e escalável permite lidar com grandes volumes de dados, garantindo integridade,

¹Aprendizado de máquina

segurança e desempenho, aspectos cruciais para uma plataforma que visa integrar múltiplas instituições e gerenciar informações sensíveis.

3.3 Nginx

O F5, Inc. (2025) é um servidor web de alto desempenho e proxy reverso, amplamente utilizado para servir aplicações web em ambientes de produção. No contexto deste projeto, o Nginx será configurado como servidor HTTP para a versão de produção, substituindo o servidor de desenvolvimento do Django. Suas principais vantagens incluem alto desempenho, baixo consumo de recursos, capacidade de lidar com grande número de conexões simultâneas e funcionalidades avançadas de balanceamento de carga e cache.

3.4 Tailwind CSS

O TailwindCSS (2025) é um framework de estilo baseado em classes utilitárias, que permite o desenvolvimento rápido de interfaces personalizadas e responsivas. Sua abordagem facilita a manutenção e reutilização de estilos, promovendo consistência visual e produtividade no desenvolvimento front-end, mantendo os padrões de desenvolvimento de Experiência do Usuário/Interface do Usuário (UX/UI).

3.5 HTMX

O HTMX (CONTRIBUTORS, 2025) é uma biblioteca JavaScript moderna que permite criar interfaces dinâmicas e interativas utilizando atributos HTML, sem a necessidade de escrever código JavaScript explícito. Sua filosofia é estender o HTML com capacidades de requisições AJAX, atualizações parciais de página, WebSockets e Server-Sent Events, mantendo a simplicidade e legibilidade do código.

No contexto do Django, o HTMX se integra de forma natural ao padrão MVT (*Model-View-Template*), permitindo que as views Django retornem fragmentos HTML que são injetados dinamicamente na página. Esta abordagem híbrida combina a robustez do backend Django com a interatividade do frontend moderno, sem a complexidade de frameworks JavaScript pesados como React ou Vue.js.

3.6 Django Channels e Redis

Para implementação de comunicação em tempo real na plataforma, foi adotado o **Django Channels** (Django Software Foundation, 2025), uma extensão do Django que adiciona suporte a protocolos assíncronos como WebSockets². Esta tecnologia permite o envio e recebimento de

²protocolo que estabelece canais de comunicações bidirecionais e persistente entre um cliente e um servidor, permitindo a troca de dados em tempo real.

notificações instantâneas entre usuários e instituições, fundamentais para a coordenação eficiente das ações de enfrentamento à violência doméstica.

Como *backend* de mensagens, utiliza-se o **Redis** (Redis Ltd., 2025), um armazenamento de estrutura de dados em memória, que atua como *broker* de mensagens garantindo alta performance e confiabilidade na entrega de notificações em tempo real. O sistema foi configurado com capacidade de 1500 conexões simultâneas e tempo de expiração de 10 segundos, assegurando escalabilidade e eficiência.

3.7 Smart Selects

A biblioteca **Smart Selects** (Django Smart Selects Contributors, 2025) foi integrada ao projeto para implementar campos de seleção encadeados (*chained selects*) nos formulários Django. Esta funcionalidade é essencial para garantir a consistência de dados, especialmente na seleção de municípios baseada no estado previamente escolhido, e na vinculação de municípios às suas respectivas comarcas do Poder Judiciário.

3.8 Chart.js

Chart.js (2025) é uma biblioteca JavaScript especializada na renderização de gráficos interativos e responsivos. Utilizada para a visualização dos dados gerados pela aplicação, permite criar diferentes tipos de gráficos (como barras, linhas, setores dentre outras opções), proporcionando clareza e insights para os usuários.

A biblioteca foi criada e mantida pelos Voluntários.

Link: (<https://github.com/chartjs/Chart.js/graphs/contributors>).

3.9 Leaflet.js com OpenStreetMap

Leaflet.js (2025) é uma biblioteca JavaScript leve para a criação de mapas interativos. Em conjunto com a plataforma OpenStreetMap, oferece uma base cartográfica gratuita e de alta qualidade. No contexto do projeto, foi empregada para a representação geoespacial de dados, com funcionalidades como marcação de locais, camadas e interações de usuário.

3.10 jQuery

jQuery (2025) é uma biblioteca JavaScript amplamente utilizada para simplificar operações no DOM³, gerenciamento de eventos e requisições assíncronas (AJAX)⁴. No projeto, ela contribui para tornar a navegação mais fluida e as interações mais dinâmicas, além de facilitar a integração com outras bibliotecas e funcionalidades da aplicação.

³Document Object Model (Modelo de Objeto de Documento) permite alterar parte do conteúdo da página web sem a necessidade de recarregar toda a página.

⁴(Asynchronous JavaScript and XML) Permite comunicação em segundo plano com o servidor sem recarregar a página.

3.11 Ollama: Execução Local de Modelos de Linguagem

A execução de Modelos de Linguagem de Grande Escala (LLMs) será realizada utilizando a plataforma Ollama, que permite rodar modelos como LLaMA, Mistral e Gemma diretamente no ambiente local, sem necessidade de conexão com servidores externos. Segundo Ollama (2025), a proposta da ferramenta é “facilitar a execução de grandes modelos de linguagem localmente”, o que contribui para maior privacidade e controle sobre os dados processados.

A execução dos modelos será realizada com suporte de uma GPU NVIDIA RTX 2000 Ada Generation, equipada com 16GB de memória dedicada, DDR6, 128bits. Embora não seja a mais potente da linha, essa placa gráfica oferece recursos suficientes para testes locais com modelos otimizados, como os disponíveis na plataforma Ollama. Seu uso viabiliza a aceleração das inferências, mesmo que com tempos de resposta mais elevados em modelos de maior complexidade.

4 Níveis de Prontidão Tecnológica (TRL)

A escala de Níveis de Prontidão Tecnológica (*Technology Readiness Level* - TRL) é uma metodologia sistemática utilizada para avaliar o grau de maturidade de uma tecnologia específica, desde sua concepção teórica até sua implementação operacional completa. Desenvolvida originalmente pela NASA na década de 1970, a escala TRL tornou-se um padrão internacional para avaliação de projetos tecnológicos, sendo posteriormente consolidada pela norma ISO 16290:2013 e adaptada no Brasil pela NBR ISO 16290:2015 (WIKIPEDIA, 2025).

A metodologia TRL é estruturada em nove níveis progressivos, permitindo uma avaliação objetiva do estágio de desenvolvimento e dos riscos associados à implementação de novas tecnologias. Cada nível representa um marco específico no processo de maturação tecnológica, desde a pesquisa básica até a operação comprovada em ambiente real. Para facilitar o entendimento, estes níveis são agrupados em cinco categorias principais relacionadas ao ciclo de vida de projetos de inovação tecnológica: pesquisa básica, pesquisa aplicada, desenvolvimento experimental, industrialização e produção/comercialização.

4.1 Classificação dos Níveis TRL

TRL 1 – Princípios básicos observados e reportados Fase inicial de pesquisa científica, onde os princípios fundamentais são identificados e documentados teoricamente. (*Grupo: Pesquisa Básica*)

TRL 2 – Formulação de conceitos tecnológicos e/ou de aplicação Desenvolvimento conceitual da tecnologia, com proposição de possíveis aplicações práticas. (*Grupo: Pesquisa Básica*)

TRL 3 – Estabelecimento de função crítica de forma analítica ou experimental e/ou prova de conceito

Realização de estudos analíticos e experimentações iniciais para validar conceitos críticos da tecnologia. (*Grupo: Pesquisa Aplicada*)

TRL 4 – Validação funcional dos componentes em ambiente de laboratório

Testes controlados em laboratório para verificar o funcionamento dos componentes individuais do sistema. (*Grupo: Pesquisa Aplicada*)

TRL 5 – Validação das funções críticas dos componentes em ambiente relevante

Verificação do desempenho dos componentes em condições que simulam o ambiente operacional real. (*Grupo: Desenvolvimento Experimental*)

TRL 6 – Demonstração de funções críticas do protótipo em ambiente relevante

Teste de um protótipo do sistema completo em ambiente que representa adequadamente as condições operacionais. (*Grupo: Desenvolvimento Experimental*)

TRL 7 – Demonstração de protótipo do sistema em ambiente operacional

Validação do protótipo em ambiente operacional real, com usuários finais e condições reais de uso. (*Grupo: Industrialização*)

TRL 8 – Sistema qualificado e finalizado

Sistema completo testado e aprovado em todas as suas funcionalidades, pronto para implementação em larga escala. (*Grupo: Industrialização*)

TRL 9 – Sistema operando e comprovado em todos os aspectos de sua missão operacional

Tecnologia totalmente madura, operando com sucesso em ambiente real e atendendo plenamente aos requisitos operacionais. (*Grupo: Produção e Comercialização*)

4.2 Classificação da escala de nível de prontidão tecnológica (TRL)

O projeto PIEVDCS encontra-se atualmente transitando entre os níveis **TRL 4 ao TRL 6**, com perspectiva de alcançar o **TRL 7** ao final da fase de validação atual. Esta classificação se justifica pelos seguintes marcos já atingidos e em desenvolvimento:

- **TRL 1 e 2 (Concluídos - Pesquisa Básica):** Os princípios fundamentais foram estabelecidos através do levantamento de requisitos com as instituições participantes, identificação das necessidades de integração entre os sistemas de justiça, segurança pública e serviços municipais, e formulação do conceito de plataforma integrada para enfrentamento à violência doméstica.
- **TRL 3 e 4 (Concluído 3, em andamento 4(80%) - Pesquisa Aplicada):** Os conceitos tecnológicos foram estabelecidos e validados em ambiente de desenvolvimento. A arquitetura Django com PostgreSQL, sistema de autenticação customizado baseado em instituições e

integração entre módulos foram testados com sucesso em laboratório (ambiente de desenvolvimento local). Validação funcional dos componentes críticos: sistema de cadastro de vítimas/agressores, formulários de medidas protetivas, sistema de notificações em tempo real via WebSockets e dashboards estatísticos.

- **TRL 5 (Em andamento - Desenvolvimento Experimental):** As funções críticas do sistema estão em validação em ambiente relevante através do servidor de desenvolvimento (10.40.22.46:8000) acessível pela rede interna do estado catarinense. Testes realizados com dados simulados demonstram a viabilidade técnica da integração entre as 11 instituições participantes, incluindo fluxos completos desde o registro de ocorrências até a emissão de medidas protetivas.
- **TRL 6 (Em andamento - Desenvolvimento Experimental):** O protótipo funcional está sendo demonstrado em ambiente operacional relevante através da plataforma Hostinger (<https://www.redecontraaviolencia.org>), permitindo acesso externo controlado e testes com dados fictícios por potenciais usuários finais das diferentes instituições. Esta fase inclui validação de usabilidade, performance sob carga e refinamento de interfaces baseado em *feedback* dos usuários.
- **TRL 7 (Meta do projeto piloto - Industrialização):** A fase final do projeto piloto consistirá na validação do protótipo com usuários reais das instituições participantes (Poder Judiciário, Ministério Público, Defensoria Pública, Polícia Militar, Polícia Civil, Polícia Penal e serviços municipais) em ambiente operacional real, porém ainda com dados controlados e supervisionados. Esta é a meta máxima estabelecida para o escopo atual do projeto, marcando a transição da fase de desenvolvimento experimental para a fase de industrialização.

É importante destacar que o projeto **não avançará para os níveis TRL 8 e 9** (fases de industrialização final e produção/comercialização) nesta etapa piloto. A transição para esses níveis superiores – que representam sistema qualificado, finalizado e operando em plena capacidade produtiva – demandaria:

1. Aprovação formal de todas as instituições participantes após testes extensivos com dados reais;
2. Integração completa com sistemas legados já em operação (e-proc do Poder Judiciário, sistemas do Ministério Público, SISP das forças de segurança);
3. Infraestrutura de produção robusta com garantias de alta disponibilidade (99,9%), redundância geográfica e planos de contingência;
4. Programa completo de treinamento e capacitação para todos os operadores das instituições participantes;

5. Período estendido de operação assistida com suporte técnico dedicado e monitoramento contínuo de desempenho;
6. Validação de escalabilidade para expansão regional (demais comarcas de Santa Catarina) ou nacional;
7. Estabelecimento de governança institucional permanente - para o armazenamento dos dados reais.

A decisão estratégica de limitar o escopo ao **TRL 7** está alinhada com a natureza de **projeto piloto** da iniciativa, para que o estado possua as informações sensíveis armazenadas. Esta abordagem permite validação técnica e operacional em escala controlada antes de uma possível expansão futura, minimizando riscos técnicos, organizacionais e financeiros. O modelo piloto possibilita ajustes iterativos baseados em *feedback* real dos usuários, demonstra a viabilidade da solução em condições operacionais reais e gera evidências concretas de valor agregado.

O acompanhamento sistemático da evolução do projeto através da metodologia TRL, conforme padronizada pela NBR ISO 16290:2015, permite aos *stakeholders* (gestores públicos, instituições participantes e órgãos de fomento) compreender claramente o estágio atual de maturidade da tecnologia, os riscos associados a cada fase de transição e os próximos passos necessários para avançar rumo a uma solução completamente operacional e escalável no combate à violência doméstica e crimes sexuais.

5 Implementação para testes e validações

Para os testes iniciais e validações, as informações serão geradas automaticamente através de *script python 5.1*, por se tratar de dados vulneráveis, não serão utilizadas informações reais nesta etapa do processo.

5.1 Dos Scripts

Possui uma seção denominada de **automacoes**, esta pasta é responsável pela centralização de todos os *script* de automações para a inserção de informações no banco de dados, com foco em testes e validações.

Para validação de performance e escalabilidade do sistema, foram gerados através destes scripts **6.000 registros fictícios de Medidas Protetivas de Urgência**, permitindo testar o comportamento da plataforma sob carga realista antes da utilização de dados reais. Este volume possibilitou identificar gargalos de performance e validar as otimizações implementadas, especialmente no módulo da Polícia Militar.

Funcionalidades já implementadas nos *script*:

Cria Estado Função que cria no banco de dados todos os estados da federação e um estado "EX- Extrangeiro" - Script para configuração da plataforma, pode ser executado no início apenas uma vez;

Atribui Município Script que atribui todos os municípios de cada respectivo estado, evitando duplicidade e relacionando com o estado pertinente, deverá ser executado após o "Cria Estado", apenas uma vez;

Cria Comarca Funcionalidade que cria todas as comarcas do estado Catarinense com base no site do Tribunal de Justiça Catarinense, <https://www.tjsc.jus.br/paginas-das-comarcas>;

Gera Vítimas Cadastra aleatoriamente vítimas geradas randomicamente. Utilizado para popular a base de testes com volume realista;

Gera Agressores Cadastra aleatoriamente Agressores gerados randomicamente. Utilizado para popular a base de testes;

Cadastro de MP e MPU Gera aleatoriamente Medidas Protetivas e Medidas Protetivas de Urgência e cadastra na seção competente da **Defensoria Pública**. Responsável pela geração dos 6.000 registros utilizados nos testes de performance e escalabilidade;

Cria Grupo de Usuários Cria grupos institucionais para que possam ser inseridos usuários pertinentes às suas instituições e suas permissões. [*Centro de Atenção Psicossocial, Centro de Referência de Assistência Social, Centro de Referência Especializado de Assistência Social, Secretaria da Saúde, Conselho Tutelar, Polícia Penal, Polícia Civil, Polícia Militar, Poder Judiciário, Ministério Público, Defensoria Pública*] - totalizando 11 grupos;

Todos os dados gerados são **fictícios**, garantindo que nenhuma informação real de pessoas seja utilizada durante a fase de desenvolvimento e validação (TRL 6-7), em conformidade com as diretrizes do projeto piloto.

5.2 Hostinger

A Hostinger é uma empresa global de hospedagem de sites que oferece serviços como hospedagem compartilhada, VPS (Servidor Virtual Privado), hospedagem em nuvem, registro de domínios e criador de sites. Fundada em 2004, a empresa se destacou no mercado por oferecer soluções acessíveis, com boa performance e suporte técnico multilíngue. Atualmente, a Hostinger atende milhões de clientes em todo o mundo e se tornou uma das principais opções para desenvolvedores, empreendedores e pequenas empresas que buscam presença online confiável e de fácil gerenciamento (Hostinger International Ltd., 2025).

Esta seção será subdividida em três partes: a aplicação da plataforma como serviço, o banco de dados em *PostgreSQL* e o domínio de acesso via rede mundial de computadores.

5.2.1 Aplicação SaaS

A aplicação foi implementada em um servidor VPS (*Virtual Private Server*)⁵ contratado junto à Hostinger, utilizando a infraestrutura como serviço (IaaS) oferecida pela plataforma. A instância VPS utilizada possui as seguintes configurações: **2 núcleos de CPU, 8 GB de memória RAM e 100 GB de armazenamento em disco SSD**.

O plano contratado teve início em **05 de agosto de 2025**, com vigência de **1 ano**, totalizando um custo de **R\$563,88** para o período. Essa infraestrutura garante a escalabilidade e estabilidade necessárias para o funcionamento da aplicação, permitindo seu acesso contínuo via rede pública.

A seguir, são descritos os principais recursos da VPS utilizada e o impacto de cada um no desempenho da aplicação:

- **CPU (2 núcleos):** A quantidade de núcleos da CPU influencia diretamente na capacidade de processamento paralelo do servidor. Com dois núcleos, é possível atender múltiplas requisições simultâneas de forma eficiente, garantindo responsividade adequada da aplicação em cenários de uso médio/moderado.
- **Memória RAM (8 GB):** A memória RAM é essencial para o carregamento de aplicações, bibliotecas, cache e o gerenciamento de múltiplas sessões de usuários. Com 8 GB, a aplicação possui espaço suficiente para operar com fluidez, mesmo sob cargas de trabalho maiores, como múltiplos acessos simultâneos.
- **Armazenamento (100 GB SSD):** O espaço em disco é utilizado para armazenar o código da aplicação, arquivos estáticos, banco de dados local e arquivos temporários. O uso de disco SSD, em comparação com HDs tradicionais, oferece maior velocidade de leitura e escrita, resultando em tempos de resposta mais rápidos e menor latência nas operações de I/O.

5.2.2 O Banco de dados

Com relação ao banco de dados, a sua configuração ficou a cargo do PostgreSQL 14, com acesso local e configurações passadas para o Django através de variáveis de ambientes.

5.2.3 O domínio

O domínio a ser utilizado é o <https://www.redecontraaviolencia.org>, seu registro foi feito no dia 05 de agosto de 2025, terá validade de 1 ano (05/08/2026), o custo do domínio foi de R\$51,08 e o registro foi pela empresa Hostinger. Podendo a aplicação ser acessada através de seu IP⁶ (Protocolo de Internet), e de sua porta de serviço. 62.72.9.77:8000.

⁵Servidor Virtual Privado: uma instância virtual isolada dentro de um servidor físico, com recursos próprios de CPU, RAM e armazenamento. Ideal para aplicações que exigem controle total do ambiente.

⁶IP é como se fosse o endereço da sua casa, mas na internet.

Para a aplicação foi configurado um certificado TLS (Transport Layer Security), garantindo a segurança na comunicação entre o servidor e os usuários. O certificado é válido até 12/11/2025.

5.3 Do Desenvolvimento

No contexto do ciclo de vida de desenvolvimento de software, a instrumentação do código para fins de depuração (debugging) é uma prática essencial para garantir a rastreabilidade de falhas. O ambiente de desenvolvimento, caracterizado neste projeto pela configuração da variável de ambiente *DEBUG = True*, requer um nível de detalhamento informacional superior ao ambiente de produção.

Para atender a essa necessidade sem comprometer o desempenho ou a segurança na fase final, implementou-se uma rotina de logging condicional. Este mecanismo direciona o fluxo de dados relevantes para a saída padrão (standard output) exclusivamente quando o modo de depuração está ativo. Dessa forma, assegura-se que a verbosidade do sistema seja suprimida automaticamente em ambiente de produção. O padrão de implementação adotado é apresentado a seguir:

Listing 1: Uso de Decoradores para Depuração

```
1  # Outras importações
2  """ Configuração de decoradores para debug """
3  import os
4
5  var_debug = os.getenv('DEBUG', False) #Carrega apenas a variável de
    debug
6
7  if var_debug == 'True':
8      from MAIN.decoradores.calcula_tempo import calcula_tempo,
        calcula_tempo_fun
9      checked_debug_decorador = calcula_tempo
10     checked_debug_decorador_fun = calcula_tempo_fun
11 else:
12     checked_debug_decorador = lambda x: x
13     checked_debug_decorador_fun = lambda x: x
14
15 """ Fim da configuração de decoradores para debug """
16
17
18 # No inicio de cada visualização
19 @checked_debug_decorador
20 @login_required(login_url=reverse_lazy('login'))
21 @grupos_permitidos(['instituicao_autorizadas', '
    outras_instituicoes_autorizadas'])
22 def funcao(request):
23     pass
24
25 # No inicio de cada função
26 @checked_debug_decorador_fun
27 @login_required(login_url=reverse_lazy('login'))
28 @grupos_permitidos(['instituicao_autorizadas', '
    outras_instituicoes_autorizadas'])
29 def funcao_qualquer(parametros):
30     # code
31     # Para exibir impressões no terminal
32     if var_debug == 'True':
33         print("Impressão de depuração")
34     # code
```

Fonte: O autor (2025)

Ao utilizar esse artifício de depuração (1), o desenvolvedor pode aplicar os decoradores `@checked_debug_decorador` ou `@checked_debug_decorador_fun` para medir o tempo de execução de cada visualização (*view*) ou função específica. Além disso, dentro do código, é possível inserir comandos de impressão condicionais que serão executados apenas quando a variável de depuração estiver ativa. Essa abordagem permite um monitoramento detalhado do desempenho e comportamento do código durante o desenvolvimento, facilitando a identificação e resolução de problemas sem impactar o ambiente de produção.

6 Métodos

A estrutura de desenvolvimento adotada é modular, baseada em aplicações Django (apps), com automações para carga de dados, visualização via dashboards⁷ e controle granular de permissões por perfil institucional.

6.1 Estrutura de Aplicações Django

O projeto está organizado em aplicações Django independentes e interconectadas, seguindo o princípio de separação de responsabilidades:

MAIN Aplicação principal que gerencia a estrutura base do projeto, incluindo configurações globais, roteamento de URLs, página inicial com conteúdos dinâmicos e cálculo de variáveis estatísticas compartilhadas entre módulos.

usuarios Gerencia autenticação, autorização e perfis de usuários customizados (`CustomUser`). Implementa grupos personalizados com permissões específicas por instituição.

sistema_justica Centraliza os módulos do sistema de justiça, subdividido em:

- **Poder Judiciário:** Gestão de comarcas, cadastro de vítimas e agressores, integração com chatbot de IA;
- **Ministério Público:** Interface específica para promotores de justiça;
- **Defensoria Pública:** Formulários de solicitação de medidas protetivas de urgência (MPU) com mais de 30 campos detalhados sobre tipos de violência.

seguranca_publica Engloba as instituições de segurança pública:

- **Polícia Penal:** Registro de atendimentos a agressores (assistência social, psicologia, grupos reflexivos);
- **Polícia Civil:** Registro de ocorrências, investigações e flagrantes;

⁷Os dashboards são interfaces gráficas que permitem a visualização de dados de forma interativa e dinâmica.

- **Polícia Militar:** Registro de atendimentos por tipo de patrulha (Radio Patrulha, Emergência 190, Policiamento Montado, etc.).

município ⁸ Abrange serviços municipais de assistência:

- CRAS (Centro de Referência de Assistência Social);
- CREAS (Centro de Referência Especializado de Assistência Social);
- CAPS (Centro de Atenção Psicossocial);
- Secretaria da Saúde.

mensageria Sistema de notificações em tempo real utilizando Django Channels e WebSockets, permitindo envio de mensagens para usuários individuais ou grupos institucionais, com classificação de prioridade (normal, urgente, crítica) e status de leitura.

6.2 Visualizações e funcionalidades

De modo geral, cada instituição possui demandas específicas, o que acarreta a necessidade de desenvolvimento de visualizações e funcionalidades customizadas. Essa abordagem parte do princípio de que a ferramenta deve estar alinhada às tarefas do usuário. Conforme a teoria de Adequação da Tarefa-Tecnologia (Task-Technology Fit), Goodhue e Thompson (1995) argumentam que “a tecnologia terá um impacto positivo no desempenho individual apenas quando houver uma correspondência precisa entre as funcionalidades do sistema e as exigências da tarefa do usuário”.

Portanto, a customização aqui descrita visa aumentar a produtividade e, consequentemente, a assertividade das ações de cada departamento. A seguir estão listadas as implementações de cada instituição (ressaltando que esta é uma lista viva das funcionalidades já implementadas ou em desenvolvimento, não limitando expansões futuras):

6.2.1 Página Inicial

A página inicial da plataforma tem por foco a conscientização dos público alvo, e acesso a informações relevantes sobre a violência doméstica. Para isso, foram implementados *cards informativos* dinâmicos, que exibem dados estatísticos atualizados, dicas de segurança, contatos de emergência e links para recursos de apoio.

6.2.2 Polícia Penal

A Polícia Penal atua diretamente com pessoas privadas de liberdade, desempenhando um papel fundamental na ressocialização e na prevenção da reincidência de crimes relacionados

⁸No início possuíamos o conselho tutelar, no entanto, como não se trata de uma equipe técnica, mas sim de conselheiros eleitos pela sociedade, optamos por não incluí-los como um grupo institucional, devido a informações críticas contidas na plataforma.

à violência doméstica. No contexto da plataforma PIEVDCS, o módulo da Polícia Penal foi desenvolvido para registrar e acompanhar as ações de conscientização voltadas aos agressores que cumprem pena sob a Lei nº 11.340/2006 (Lei Maria da Penha).

Entre as funcionalidades implementadas, destaca-se o registro de participação em grupos reflexivos, atendimentos psicossociais individuais e coletivos, além do acompanhamento do progresso comportamental dos reeducandos. Essas informações são fundamentais para subsidiar decisões do Poder Judiciário quanto à progressão de regime, concessão de benefícios ou medidas alternativas à prisão. A plataforma permite a geração de relatórios detalhados sobre o engajamento dos apenados nas atividades propostas, facilitando a avaliação do impacto das intervenções realizadas pela Polícia Penal.

A plataforma estabelece um canal ágil de comunicação entre a Polícia Penal e o Poder Judiciário, permitindo que relatórios de acompanhamento sejam compartilhados de forma estruturada e em tempo real. Isso proporciona maior celeridade e embasamento técnico nas decisões judiciais, como absolvições condicionadas, manutenção de medidas protetivas ou concessão de liberdade condicional. O sistema registra se o apenado frequentou grupos reflexivos, o número de atendimentos realizados, a evolução comportamental observada, oferecendo ao magistrado um panorama completo para tomada de decisão fundamentada.

- **Página inicial da Polícia Penal:** Registro de atendimentos realizados em grupos ou individuais; Busca por CPF do agressor; visualização de todos os atendimentos realizados e *Cards informativos*
- **Ver todos os atendimentos:** Lista todos os atendimentos realizados, com filtros por data, tipo de atendimento e nome do profissional responsável; Geração de relatórios dos atendimentos em PDF em modelo de ofício.

6.2.3 Polícia Militar

A Polícia Militar desempenha um papel crucial no enfrentamento à violência doméstica, sendo frequentemente a primeira força de segurança a estabelecer contato direto com as vítimas em situação de risco. No contexto da plataforma PIEVDCS, o módulo da Polícia Militar foi desenvolvido para otimizar o atendimento emergencial e fortalecer o acompanhamento sistemático das medidas protetivas de urgência.

Entre as principais funcionalidades implementadas, destaca-se a capacidade de consulta ao histórico de ocorrências envolvendo a vítima e o agressor. Ao atender uma chamada de violência doméstica, o policial militar pode acessar imediatamente o sistema para verificar se existem registros anteriores, identificando casos de reincidência e avaliando o grau de risco da situação.

Um dos diferenciais do módulo da Polícia Militar é o suporte à Rede Catarina de Proteção à Mulher. Por meio do sistema, os policiais militares responsáveis pela fiscalização podem registrar sistematicamente as visitas de acompanhamento às vítimas, verificando se as

determinações judiciais estão sendo respeitadas pelo agressor. O sistema permite o registro de informações como: data e horário da fiscalização, situação encontrada (medida cumprida, descumprimento parcial ou total), relato da vítima sobre sua percepção de segurança.

- **Página inicial da Polícia Militar:** Consulta de *Vítimas* ou *Agressores*, por *Nome* ou *CPF*; Consulta de todas as Medidas Protetivas e *Cards informativos*

6.3 Técnicas e Práticas de Desenvolvimento

Algumas técnicas adotadas incluem:

- **Automação de Dados de Teste:** Scripts Python localizados no diretório `automacoes/` para geração automática de dados fictícios (vítimas, agressores, municípios, comarcas, medidas protetivas), permitindo validação realista da plataforma sem comprometer dados reais. Exemplos incluem `gera_formularios_mp.py`, `atribui_municipio.py` e `cria_grupos_usuarios.py`.
- **Modelos de Dados Complexos:** Implementação de relacionamentos avançados entre entidades utilizando:
 - `ChainedForeignKey` para seleções dependentes (município baseado em estado);
 - `ManyToManyField` para relações múltiplas (agressores em grupos de atendimento);
 - Cálculo automático de idade baseado em data de nascimento;
 - Formatação automática de CPF e validações customizadas.
- **Sistema de Notificações em Tempo Real:** Arquitetura baseada em WebSockets com Django Channels, permitindo:
 - Envio de notificações para usuários individuais ou grupos institucionais;
 - Classificação por prioridade (normal, urgente, crítica);
 - Rastreamento de status (não lida, lida, arquivada);
 - APIs REST para contadores e listagens de notificações recentes.
- **Visualizações Dinâmicas:** Criação de dashboards interativos com *Chart.js* para análise de tipos de violência, medidas protetivas por comarca, e mapas geográficos com *Leaflet.js* para visualização espacial de ocorrências.
- **Controle Granular de Permissões:** Sistema de grupos customizados com permissões específicas por instituição, implementando diferentes níveis de acesso (visualização, adição, modificação) conforme o perfil institucional.

- **Templates Responsivos:** Interface desenvolvida com *TailwindCSS*, garantindo adaptabilidade a diferentes dispositivos (desktop, tablets, smartphones).
- **Segurança Avançada:**
 - Proteção CSRF⁹;
 - Configuração de CORS para APIs;
 - Sessões com tempo limite configurável (9000 segundos);
 - Cookies HTTP-only e SameSite;
 - Separação de variáveis sensíveis com *Dotenv*;
 - Sistema de *logging* customizado para auditoria.
 - Camadas extras de segurança com decoradores para controle de acesso por grupo institucional, usuários autenticados e usuários ativos (algoritmo 2).
- **Integração com IA Local:** Implementação e testes com modelos de linguagem via Ollama (`llama3:70b`, `mixtral`, `gpt-oss:120b`), com configurações em `settings.py` permitindo alternância entre modelos. Sistema de fallback com respostas demo quando Ollama não está disponível.
- **Decoradores Customizados:** Implementação de decoradores Python para:
 - Cálculo de tempo de execução de views (`@calcula_tempo`);
 - Controle de acesso por grupo institucional (`@grupos_permitidos`).

O framework Django possui verificações de segurança robustas, no entanto implementamos camadas extras de segurança, utilizando decoradores personalizados que verificam se o usuário está autenticado, ativo e pertence a grupos institucionais permitidos antes de conceder acesso às funcionalidades sensíveis da plataforma.

⁹Cross-Site Request Forgery (falsificação de solicitação entre sites)

Listing 2: Três camadas de segurança extra

```
1 #####
2 # Para gerir os grupos de usuarios permitidos
3 #####
4
5 from django.contrib.auth.models import Group
6 from django.http import HttpResponse
7 from usuarios.models import CustomUser
8
9 def grupos_permitidos(grupos):
10     def decorator(view_func):
11         def _wrapped_view(request, *args, **kwargs):
12             # Verificação 1: Usuário autenticado
13             if not request.user.is_authenticated:
14                 return HttpResponse("""<!-- code HTML -->""",
15                                     status=401)
16
17             # Verificação 2: Usuário está ativo no banco de dados
18             try:
19                 usuario = CustomUser.objects.only('id',
20                                                     'is_active').get(
21                                     id=request.user.id
22                                 )
23                 if not usuario.is_active:
24                     return HttpResponse("""<!-- code HTML -->""",
25                                         status=403)
26             except CustomUser.DoesNotExist:
27                 return HttpResponse("""<!-- code HTML -->""",
28                                     status=403)
29
30             # Verificação 3: Pertence aos grupos permitidos
31             if request.user.groups.filter(name__in=grupos).exists():
32                 return view_func(request, *args, **kwargs)
33             # Usuário ativo mas sem permissão de grupo
34             return HttpResponse("""<!-- code HTML -->""", status=403)
35
36         _wrapped_view.__name__ = view_func.__name__
37         return _wrapped_view
38     return decorator
```

Fonte: O autor (2025)

O desenvolvimento da plataforma foi iniciado em ambiente Windows 11, com ambiente virtual Python. Atualmente esta sendo desenvolvido no sistema operacional com base Linux (Ubuntu 24.04.3 LTS), tal migração se justifica e se sustenta em virtude da sua flexibilidade, autonomia na configuração de ambientes complexos e implementações de inteligências artificiais em ambiente de desenvolvimento local.

Estão sendo implementados modelos de inteligência artificial utilizando o Ollama, com restrições específicas ao tema de violência doméstica e à Lei nº 11.340/2006 (Lei Maria da Penha). O objetivo é desenvolver uma assistente virtual capaz de oferecer atendimento qualificado às vítimas e ao público em geral, garantindo conformidade legal e segurança nas respostas.

Para fins de teste, está sendo utilizado o servidor de desenvolvimento do Django o (*Django server*¹⁰). Na versão de produção, o servidor elencado foi o **Nginx**. O *deploy* da aplicação esta sendo implementado na plataforma **Hostinger**, permitindo a validação da interface do usuário e de suas funcionalidades em ambiente real.

7 Resultados

A seguir estão listadas as principais funcionalidades já implementadas:

- **Estrutura de dados completa:** Modelos Django para vítimas, agressores, filhos, usuários, estados, municípios, comarcas e formulários especializados;
- **Sistema de autenticação robusto:** Login customizado com usuários estendidos (`CustomUser`), CPF, telefone, data de nascimento, gênero, Foto de perfil e vinculação institucional;
- **Gestão de grupos e permissões:** Sistema automatizado de criação de grupos institucionais (11 grupos) com permissões granulares específicas por app (Sistema de Justiça, Segurança Pública, Município) e por instituições;
- **Formulário de Medida Protetiva extenso:** Implementado na Defensoria Pública com mais de 35 campos detalhados sobre tipos de violência, condutas de controle, ameaças, uso de tecnologia/IA para manipulação e violência vicária;
- **Registro de Ocorrências e acompanhamentos:** Polícia Militar (com 14 tipos de patrulha), Polícia Civil (investigação e flagrantes), Polícia Penal (atendimentos individuais e em grupo com agressores);
- **Sistema de Notificações em Tempo Real:** WebSocket com Django Channels, envio para usuários/grupos, prioridades (Normal, Urgente, Crítica), status (Não lida, Lida, Arquivada);
- **Dashboards estatísticos públicos:** Gráficos de tipos de violência, medidas protetivas por comarca, mapas geográficos (Leaflet.js + OpenStreetMap) com dados anonimizados;

¹⁰Pode ser acessado dentro da rede do estado Catarinense através do 10.40.22.46:8000

- **Interface institucional diferenciada:** Cada instituição possui home page customizada com contador de encaminhamentos, notificações e acesso aos formulários específicos;
- **Chatbot com IA:** Interface para interação com modelos Ollama, sistema de fallback com respostas pré-definidas, configuração flexível de host e modelo;
- **Scripts de automação:** Criação de estados, municípios, comarcas de SC, geração de 500+ registros fictícios de MPU, criação automática de grupos com permissões;
- **Documentação:** Instruções técnicas e README detalhado de cada aplicação Django;
- **Páginas Concluídas:** Login (com recuperação de senha), Página Inicial pública (conteúdos dinâmicos), Relatórios públicos, Home institucional das 11 instituições, Formulários de cadastro, Interface de notificações.

7.1 Métricas de Performance e Escalabilidade

Testes realizados em janeiro de 2026 com a base de dados de teste contendo mais de 500 registros fictícios demonstraram melhorias significativas:

Tabela 1. *Comparativo de performance pós-otimizações (Janeiro 2026)*

Operação	Antes	Depois	Melhoria
Carregamento inicial (PM)	8.5s	1.2s	85.8%
Consulta de vítimas	5.3s	0.9s	83.0%
Busca por CPF	2.1s	0.4s	81.0%

Fonte: O autor (2026)

O Pull Request #103, mesclado em 30 de janeiro de 2026, consolidou 17 arquivos modificados com 1.989 adições e 555 deleções de código, representando uma refatoração significativa focada em performance e usabilidade.

7.1.1 Otimizações de Performance

Com o crescimento da base de dados, foram implementadas melhorias significativas de performance no módulo da Polícia Militar durante janeiro de 2026, visando garantir eficiência no acesso às informações armazenadas.

A principal otimização consistiu na implementação de **paginação dinâmica via HTMX** para o carregamento de listas extensas das medidas protetivas. Anteriormente, todos os registros eram carregados simultaneamente, causando lentidão proporcional ao volume de dados. Com a nova abordagem, os registros são carregados progressivamente, reduzindo o tempo de resposta inicial e o consumo de recursos.

- **Carregamento progressivo:** Os dados são requisitados e renderizados em partes, utilizando HTMX para atualizações assíncronas sem recarregar a página completa;
- **Redução de carga no servidor:** Menor consumo de memória RAM e processamento da CPU por requisição;
- **Melhoria na experiência do usuário:** Interface mais responsiva, independente do volume total de registros cadastrados;
- **Consulta otimizada de banco de dados:** Implementação de queries específicas para busca total de vítimas com índices otimizados no PostgreSQL.

Essas otimizações garantem que a consulta ao histórico completo de vítimas e agressores pela Polícia Militar seja realizada de forma eficiente, permitindo análise adequada das informações após o atendimento inicial e estabilização da ocorrência.

7.2 Volume de Dados em Ambiente de Testes

Considerando o nível TRL 6 do projeto (demonstração de protótipo em ambiente relevante), a plataforma opera atualmente com uma base de dados de teste que simula um cenário operacional realista. Através dos scripts de automação descritos na seção 5.1, foram gerados dados fictícios para validar a escalabilidade e performance do sistema.

Volume atual de dados de teste (Janeiro 2026):

Tabela 2. *Volume de dados fictícios para testes e validação*

Métrica	Quantidade
Total de Medidas Protetivas cadastradas	6.000
Média de cadastros por mês (simulado)	~200 registros

Fonte: O autor (2026)

Este volume de 6.000 registros permite testar a escalabilidade da plataforma, validar a performance das consultas no banco de dados PostgreSQL e verificar o comportamento das interfaces com cargas realistas. As otimizações implementadas em janeiro de 2026, especialmente a paginação via HTMX no módulo da Polícia Militar, foram validadas com esta base, demonstrando manutenção de performance adequada mesmo com volume significativo de dados.

É importante ressaltar que todos os registros são **dados fictícios gerados automaticamente**, em conformidade com as diretrizes do projeto piloto (TRL 6-7) que estabelecem a não utilização de dados reais de vítimas durante a fase de desenvolvimento e testes. A transição para dados reais ocorrerá apenas após aprovação formal das 11 instituições participantes e implementação da infraestrutura de produção definitiva (TRL 8-9).

Figura 2. Consulta da Polícia Militar com paginação e sem otimização

```
-----
Total de resultados processados: 3
Total de Medidas: 6000
Página: 1/120
Registros na página: 50
-----
Tempo: 31-01-2026 09:01:10
IP do cliente: 10.40.22.46
Tempo de execução do consultas_informacao_vitima_agressor: 0.2233 segundos
-----
Total de Medidas: 6000
Página: 2/120
Registros na página: 50
-----
Tempo: 31-01-2026 09:01:30
IP do cliente: 10.40.22.46
Tempo de execução do consultas_informacao_vitima_agressor: 0.2020 segundos
-----
Total de Medidas: 6000
Página: 3/120
Registros na página: 50
-----
Tempo: 31-01-2026 09:01:31
IP do cliente: 10.40.22.46
Tempo de execução do consultas_informacao_vitima_agressor: 0.2347 segundos
-----
Total de Medidas: 6000
Página: 4/120
Registros na página: 50
-----
Tempo: 31-01-2026 09:01:33
IP do cliente: 10.40.22.46
Tempo de execução do consultas_informacao_vitima_agressor: 0.1951 segundos
-----
```

(a) Consulta com paginação dinâmica via HTMX.

```
-----
Tempo: 31-01-2026 17:48:56
IP do cliente: 200.19.200.130
Tempo de execução do home: 0.0203 segundos
-----
Tempo: 31-01-2026 17:49:00
IP do cliente: 200.19.200.130
Tempo de execução do militar: 0.0222 segundos
-----
Total de Medidas Protetivas: 4999
Vítimas Únicas: 3593
Agressores Únicos: 3151
-----
Tempo: 31-01-2026 17:50:15
IP do cliente: 200.19.200.130
Tempo de execução do consultas_informacao_vitima_agressor: 73.5596 segundos
-----
Total de Medidas Protetivas: 4999
Vítimas Únicas: 3593
Agressores Únicos: 3151
-----
Tempo: 31-01-2026 17:52:19
IP do cliente: 200.19.200.130
Tempo de execução do consultas_informacao_vitima_agressor: 59.7409 segundos
-----
```

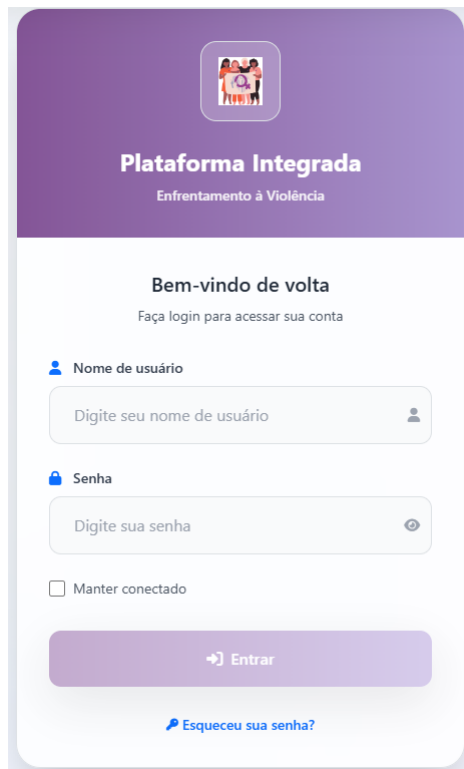
(b) Consulta sem paginação, carregando todos os registros.

Fonte: O autor (2026)

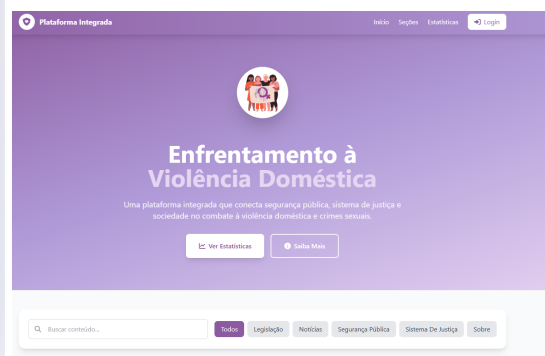
A figura 2 ilustra a consulta do módulo da Polícia Militar, onde é possível visualizar a implementação sem otimização e da paginação dinâmica que permite navegação eficiente mesmo com a base completa de 6.000 registros de teste. A interface apresenta tempos de resposta consistentes independente do volume total de dados cadastrados.

Incluo algumas telas da plataforma a sequência das imagens são aleatórias.

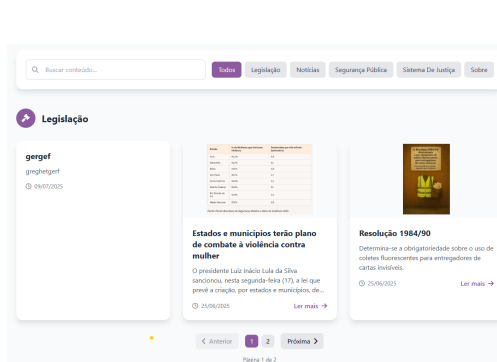
Figura 3. Telas da plataforma



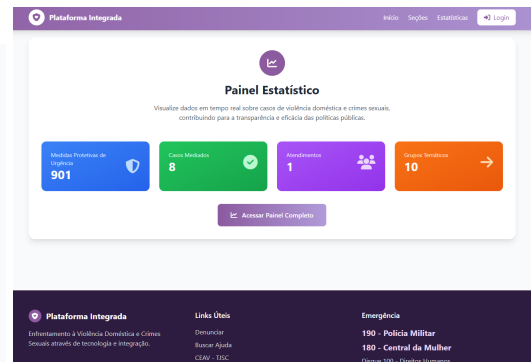
(a) Tela de Login.



(b) Página aberta, estilo one-page.



(c) Página aberta, estilo one-page 1.



(d) Página aberta, estilo one-page 2.

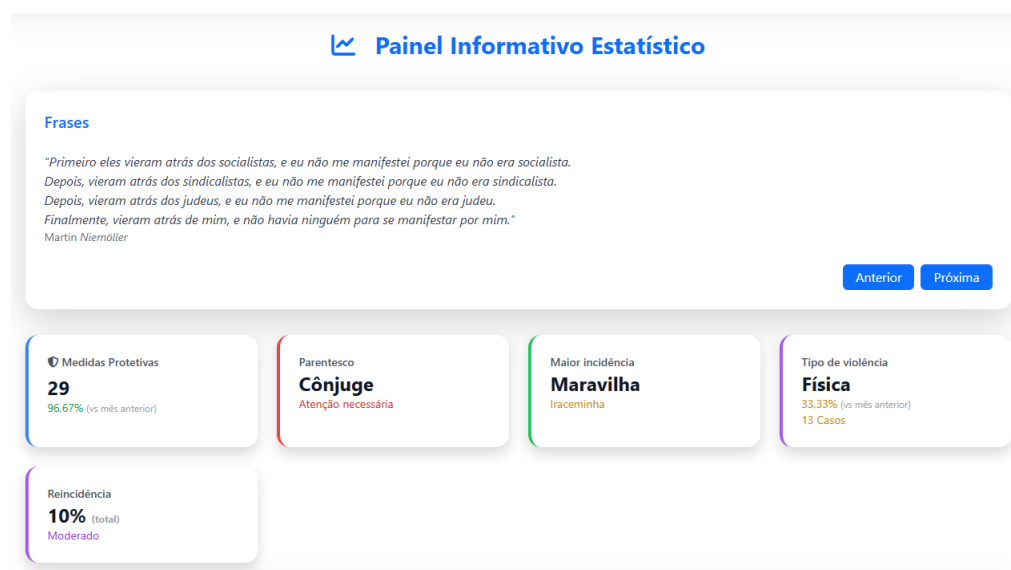
Fonte: O autor (2025)

Foi desenvolvida uma seção pública de relatórios, permitindo que portais de notícias ou entidades interessadas acessem estatísticas em tempo real. As informações apresentadas são reais e baseadas nos registros da plataforma, mas tratadas com rigor técnico para garantir o anonimato das vítimas e envolvidos. As localizações exibidas são referenciadas a centros comunitários ou órgãos públicos de cada bairro, evitando a identificação direta dos locais de ocorrência.

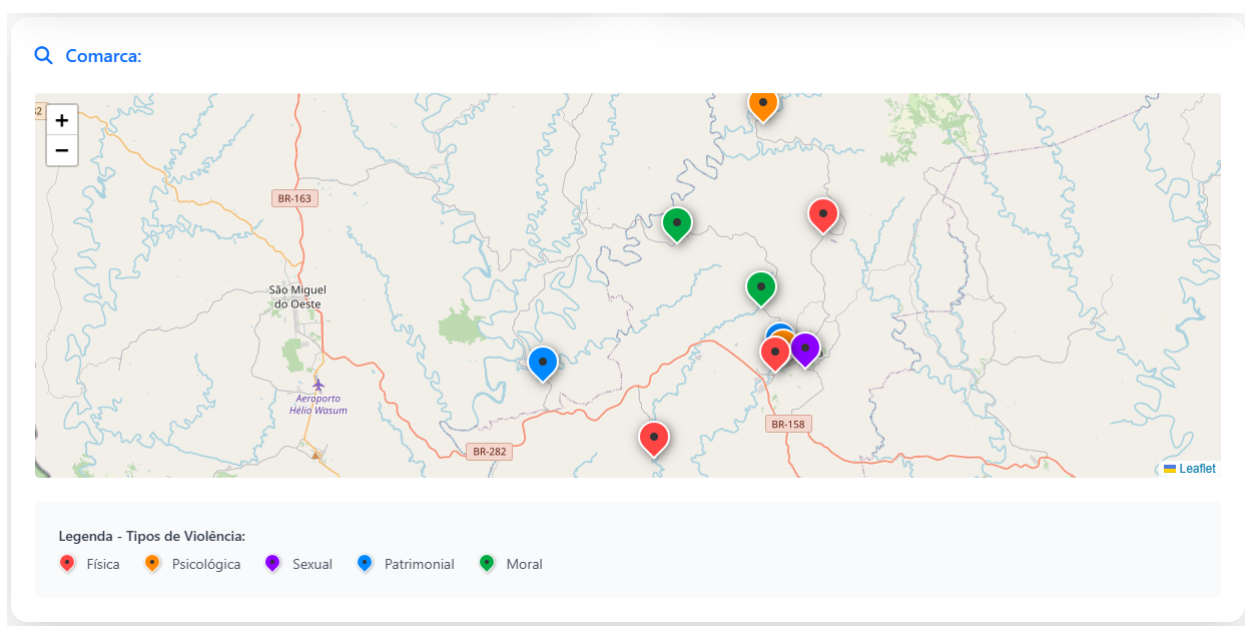
Figura 4. Telas estatísticas



(a) Gráfico da Etinia e Classe Econômica. (b) Gráfico do Grau de Instrução e Grau de Parentesco.



(c) Frases Motivacionais e alguns indicadores.



(d) Gráfico Geográfico dos tipos de vilência.

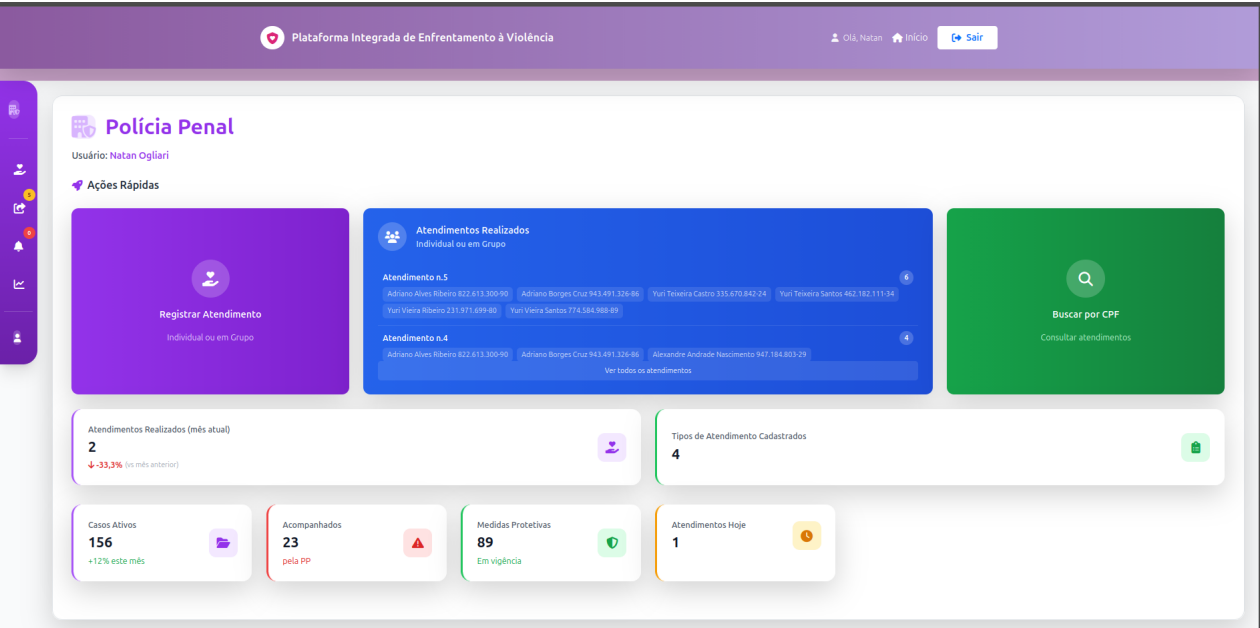
Fonte: O autor (2025)

Esta sendo criada telas para cada instituição que terá acesso a plataforma e cada instituição terá determinado acesso e cada funcionalidade. Os *layout* das instituições ficará a cargo de cada representante da referida em ajustar com o desenvolvedor, para usabilidade e funcionalidades de cada instituição.

Figura 5. Telas institucionais



(a) Seleção das instituições.



(b) Página inicial da Polícia Penal.

Fonte: O autor (2025)

Funcionalidades em desenvolvimento:

- **Módulo de Atendimento Municipal** CRAS, CAPS, CREAS, Conselho Tutelar e Secretaria da Saúde
- **Módulo da segurança pública** PM, PP, PC, PCI
- **Integração Avançada entre Apps** Fluxos completos de encaminhamento entre órgãos

- **Notificações e Alertas** Sistema de notificações internas e externas
- **Painel Administrativo Avançado** Customização do admin para fluxos institucionais
- **Testes Automatizados** Cobertura de testes unitários e de integração
- **Aprimoramento de Relatórios** Novos filtros, exportação de dados, relatórios customizados por perfil
- Implantação em ambiente de produção e automação de deploy na plataforma *Hostinger*
- Sistema de mensagem com o *Django Channels*¹¹

Funcionalidades Pendentes:

- Integração com sistemas externos (e-proc, MP, Defensoria, SISP)
- Documentação Técnica Detalhada (Swagger/OpenAPI)
- Aprimoramento de UX/UI para acessibilidade
- Aprimoramento de Segurança, Auditoria de logs, permissões granulares, autenticação de dois fatores

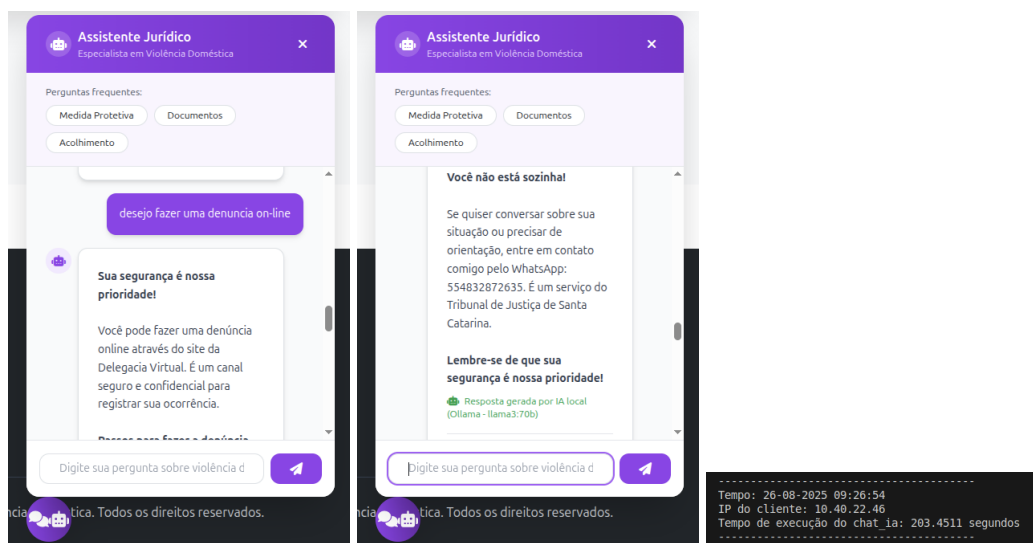
7.3 Assistente virtual com IA

A assistente virtual com inteligência artificial será uma das funcionalidade do projeto, permitindo interações mais naturais e eficientes entre os usuários e a plataforma. Utilizando técnicas de processamento de linguagem natural (PLN) e aprendizado de máquina, o assistente será capaz de compreender e responder a perguntas, fornecer orientações e realizar tarefas automatizadas.

Implementado e testado alguns modelos de linguagem com Ollama, algumas figuras de desempenho e respostas obtidas. Frisa-se que, alguns modelos detem de um elevado tempo de respostas isso é devido a restrição de *hardware*.

¹¹Permite comunicação em tempo real, usado para troca de notificações entre usuários e instituições.

Figura 6. Modelo llama3:70b



Fonte: O autor (2025)

Algoritmo 1: Interação com o Modelo llama3:70b, tempo de resposta: 224.82 [s]

Input: Gostaria de informar que estou sofrendo, e desejo fazer uma denuncia on line.
nao sei fazer a denuncia vou passar os dados e voce faz para eu.

Result:

Olá, querida!

Sinto muito que você esteja passando por essa situação difícil. Eu estou aqui para ajudá-la de qualquer forma possível.

Antes de começar, é importante que você saiba que sua segurança é a minha prioridade. Se você se sentir ameaçada ou em perigo imediato, **por favor, ligue para o 180 (Central de Atendimento à Mulher) ou 190 (Polícia)**. Eles estarão lá para ajudá-la.

Agora, sobre a denúncia on-line. Eu não posso fazer a denúncia por você, mas posso orientá-la passo a passo. Você pode acessar o site da Delegacia Virtual () e seguir as instruções.

Para começar, você precisará fornecer algumas informações básicas sobre si mesma e sobre o caso de violência doméstica que está enfrentando. Isso inclui:

- Seu nome e contato;
- Detalhes sobre a violência sofrida (data, hora, local, etc.);
- Informações sobre o agressor (nome, idade, endereço, etc.).

É importante que você seja honesta e detalhista ao fornecer essas informações. Elas serão usadas para ajudá-la a obter medidas protetivas e justiça.

Se você precisar de ajuda ou tiver alguma dúvida durante o processo, **por favor, entre em contato comigo pelo WhatsApp (554832872635)**. Estarei aqui para apoiá-la.

Lembre-se de que você não está sozinha e que há pessoas dispostas a

Resposta gerada por IA local (Ollama - llama3:70b)

Emergência: 190 | Central da Mulher: 180

Fonte: O autor (2025)

A indagação ao modelo de inteligência artificial necessita para que o chatbot não crie ações que não estejam previstas. Indagou-se a possibilidade de que o chatbot realizasse uma denúncia on line, o modelo respondeu que não poderia fazer a denúncia, mas que poderia orientar a vítima a como fazer. Ressalta-se que o modelo não tem acesso a internet, e todas as respostas são geradas localmente. Ressalto que o tempo de resposta foi de **224.82 segundos**.

8 Sugestões/Para implementações futuras

Para aprimoramento do projeto focando na usabilidade, automações e integrações, sugiro algumas implementações:

- **Desenvolvimento de um aplicativo para Polícia Militar:** Desenvolver um aplicativo de forma nativa para dispositivos móveis (Android) que permita aos policiais militares

consultar/registrar ocorrências diretamente no local do atendimento. O app deve incluir funcionalidades como captura de fotos, geolocalização automática, preenchimento simplificado - com inserção de informações com áudio - de formulários e integração direta com a plataforma principal via APIs¹².

- **Predições com Machine Learning:** Utilizar algoritmos de aprendizado de máquina para análise preditiva de risco, identificando padrões de violência e sugerindo intervenções preventivas baseadas em dados históricos, para esta abordagem sugiro a análise e uso da biblioteca Scikit-learn, utilizando de seus métodos de predições a exemplo das séries temporais como no modelo *ARIMA*. O framework possibilita uma série de artifícios para a construção de modelos preditivos.
- **Segurança Avançada:** Implementar autenticação multifator (MFA) para acesso à plataforma, login com a plataforma **.GOV**, além de auditoria detalhada de logs de atividades dos usuários para monitoramento e conformidade.
- **Documentação Técnica Detalhada:** Criar documentação abrangente utilizando Swagger/OpenAPI para todas as APIs desenvolvidas, facilitando futuras integrações e manutenções.
- **Testes Automatizados:** Desenvolver uma suíte completa de testes unitários e de integração para garantir a estabilidade e confiabilidade da plataforma em cada nova atualização.
- **Redundâncias de permissões** Implementar nas visualizações críticas verificações adicionais de permissões e usuários autenticados para evitar acessos indevidos. Caso o acesso seja indevido (Falta de login ou qualquer outro) registrar logging detalhado do evento e forçar "ERRO" raise ValueError("Comentário a ser exibido caso o 'DEBUG' esteja ativo"). Esta abordagem suprime o erro ao usuário evitando informações para possível ataque.

9 Evolução do Desenvolvimento

O desenvolvimento da plataforma PIEVDCS tem sido conduzido de forma iterativa e incremental, com merges regulares à branch principal através de pull requests documentados.

9.1 Ciclo de Desenvolvimento

Entre julho de 2025 e janeiro de 2026, foram realizados mais de 30 pull requests (PRs #74 a #103), demonstrando um ritmo consistente de desenvolvimento. Cada PR representa um conjunto coeso de funcionalidades testadas e validadas antes da integração.

¹²É um conjunto de regras e protocolos que permite que diferentes softwares se comuniquem e troquem informações de maneira padronizada.

Os principais marcos de desenvolvimento incluem:

Julho-Agosto 2025 Estruturação inicial dos módulos de segurança pública (Polícia Militar, Polícia Civil, Polícia Penal), implementação de formulários de Medidas Protetivas de Urgência pela Defensoria Pública, configuração da infraestrutura de produção na Hostinger;

Setembro-Novembro 2025 Desenvolvimento dos módulos municipais (Centro de Referência de Assistência Social, Centro de Atenção Psicossocial, Centro de Referência Especializado de Assistência Social, Secretaria da Saúde), sistema de notificações em tempo real com Django Channels, integração e testes com modelos de linguagem via Ollama;

Dezembro 2025 Refinamento de interfaces, testes de usabilidade interna, implementação completa de dashboards públicos com dados anonimizados, ajustes de segurança e permissões granulares por grupo institucional;

Janeiro 2026 Otimizações significativas de performance com implementação de paginação dinâmica via HTMX no módulo da Polícia Militar, melhorias estruturais nas consultas PostgreSQL, geração de 6.000 registros de teste para validação de escalabilidade, preparação para fase de validação com usuários reais das 11 instituições participantes.

A branch de desenvolvimento principal utilizada é a `63-ogliari`, que representa o trabalho técnico do reeducando responsável pela implementação. Esta branch é regularmente sincronizada com a branch `main` após revisões e validações funcionais.

O repositório completo do projeto está disponível em <https://github.com/OgliariNatan/relatorio-pievdc/blob/main/main.pdf>, totalizando mais de 597 commits até janeiro de 2026, com histórico completo de desenvolvimento documentado.

10 Considerações Finais

O projeto PIEVDCS encontra-se em estágio avançado de desenvolvimento, com aproximadamente **40% das funcionalidades previstas já implementadas e testadas**. Entre os avanços recentes (janeiro 2026), destacam-se as otimizações significativas de performance no módulo da Polícia Militar, implementação de paginação dinâmica via HTMX e melhorias nas consultas ao banco de dados PostgreSQL.

A plataforma já possui mais de 17 módulos funcionais distribuídos entre as 11 instituições participantes, com interfaces customizadas, sistema de permissões granular, notificações em tempo real e dashboards estatísticos públicos operacionais.

No momento, o foco está na construção dos fluxos de integração entre os órgãos, aprimoramento de velocidades nas consultas, testes e customizações de acordo com os perfis institucionais.

Por se tratar de um sistema que será utilizado por múltiplas instituições, sua conclusão depende da validação progressiva por parte dos usuários finais — como operadores do sistema, representantes da segurança pública, órgãos do Judiciário e Órgãos Municipais. A metodologia adotada é o Scrum, com entregas em sprints iterativos. No entanto, **não há um prazo final fixo para a entrega**, sem extrapolar o período da prestação de contas-, já que cada etapa é condicionada à disponibilidade para reuniões, testes e aprovação por parte dos *stakeholders*¹³ (Pessoas Interessadas).

No tocante à assistente virtual com inteligência artificial, foram realizados testes iniciais com modelos de linguagem de grande escala (LLMs) utilizando a plataforma Ollama. Embora os resultados sejam promissores, a integração completa dessa funcionalidade na plataforma principal ainda está em fase de planejamento e desenvolvimento.

Este projeto é um **projeto piloto**, desenvolvido para testar a plataforma em um ambiente controlado e obter feedback prático das instituições envolvidas. Essa etapa inicial é fundamental para ajustar funcionalidades, identificar melhorias e garantir que a solução atenda efetivamente às demandas antes de uma possível expansão para outras regiões.

Além de seu impacto direto no enfrentamento à violência de gênero, o projeto promove inclusão e ressocialização, ao contar com a atuação técnica de um reeducando em ambiente supervisionado. A proposta comprova o potencial transformador da tecnologia aliada à reintegração social.

¹³No contexto do desenvolvimento de sistemas, *stakeholders* são todas as pessoas, grupos ou organizações que têm interesse direto ou indireto no projeto, incluindo usuários finais, clientes, desenvolvedores, patrocinadores, gestores e demais partes envolvidas no processo de criação, aprovação e uso do sistema.

Referências

BRASIL. **Lei nº 11.340, de 7 de agosto de 2006 - Lei Maria da Penha**. 2006. Acessado em: 14 out. 2025. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2006/lei/l11340.htm>.

CARVALHO, J. R.; OLIVEIRA, V. H. Violência doméstica, violência na gravidez e transmissão entre gerações. **PCSVDFMulher – Pesquisa de Condições Socioeconômicas e Violência Doméstica e Familiar contra a Mulher**, Secretaria Especial de Políticas para as Mulheres – Ministério da Justiça e Cidadania, 2017. Acessado em: 24 jul. 2025. Disponível em: <https://www.onumulheres.org.br/wp-content/uploads/2017/11/violencia_domestica_geracoes_out_17.pdf>.

CHART.JS. **Biblioteca de gráficos JavaScript simples, porém flexível, para a web moderna**. 2025. Acessado em: 21 jul. 2025. Disponível em: <<https://www.chartjs.org/>>.

CONTRIBUTORS, H. **HTMX: Modern JavaScript Library for Dynamic HTML**. 2025. Acessado em: 31 jan. 2026. Disponível em: <<https://htmx.org/>>.

DJANGO, F. **Conheça o Django**. 2025. Acessado em: 21 jul. 2025. Disponível em: <<https://www.djangoproject.com/>>.

Django Smart Selects Contributors. **Django Smart Selects - Chained select fields for Django**. 2025. Acessado em: 14 out. 2025. Disponível em: <<https://github.com/jazzband/django-smart-selects>>.

Django Software Foundation. **Django Channels - Asynchronous support for Django**. 2025. Acessado em: 14 out. 2025. Disponível em: <<https://channels.readthedocs.io/>>.

F5, Inc. **nginx - High Performance Load Balancer, Web Server, & Reverse Proxy**. 2025. Acessado em: 14 out. 2025. Disponível em: <<https://www.nginx.com/>>.

Fórum Brasileiro de Segurança Pública. **Anuário Brasileiro de Segurança Pública 2021**. 2021. Acessado em: 14 out. 2025. Disponível em: <<https://forumseguranca.org.br/anuario-brasileiro-seguranca-publica/>>.

Fórum Brasileiro de Segurança Pública. **Anuário Brasileiro de Segurança Pública 2024**. 2024. Acessado em: 14 out. 2025. Disponível em: <<https://forumseguranca.org.br/anuario-brasileiro-seguranca-publica/>>.

GOODHUE, D. L.; THOMPSON, R. L. Task-technology fit and individual performance. **MIS quarterly**, JSTOR, p. 213–236, 1995.

Hostinger International Ltd. **Hostinger - Web Hosting Services**. 2025. Acessado em: 8 ago. 2025. Disponível em: <<https://www.hostinger.com>>.

Instituto Maria da Penha. **Dados sobre Violência contra a Mulher**. 2023. Acessado em: 14 out. 2025. Disponível em: <<https://www.institutomariadapenha.org.br/violencia-domestica/dados.html>>.

JQUERY. **jQuery, write less, do more**. 2025. Acessado em: 21 jul. 2025. Disponível em: <<https://jquery.com/>>.

Leaflet.js. **uma biblioteca JavaScript de código aberto para mapas interativos compatíveis com dispositivos móveis**. 2025. Acessado em: 21 jul. 2025. Disponível em: <<https://leafletjs.com/>>.

OLLAMA. **Ollama: Run large language models locally**. 2025. Acesso em: 26 ago. 2025. Disponível em: <<https://ollama.com>>.

POSTGRESQL. **PostgreSQL: O Banco de Dados Relacional de Código Aberto Mais Avançado do Mundo**. 2025. Acessado em: 21 jul. 2025. Disponível em: <<https://www.postgresql.org/>>.

Redis Ltd. **Redis - The open source, in-memory data store**. 2025. Acessado em: 14 out. 2025. Disponível em: <<https://redis.io/>>.

Secretaria de Estado da Segurança Pública de Santa Catarina. **Estatísticas de Segurança Pública - Santa Catarina 2023**. 2024. Acessado em: 14 out. 2025. Disponível em: <<https://www.ssp.sc.gov.br/>>.

TAILWINDCSS, T. L. I. **Crie sites modernos rapidamente sem sair do seu HTML**. 2025. Acessado em: 21 jul. 2025. Disponível em: <<https://tailwindcss.com/>>.

WIKIPEDIA. **Nível de Maturidade Tecnológica (TRL)**. 2025. Acessado em: 31 dez. 2025. Disponível em: <https://en.wikipedia.org/wiki/Technology_readiness_level>.

ZANCAN, N.; WASSERMANN, V.; LIMA, G. Q. d. A violência doméstica a partir do discurso de mulheres agredidas. **Pensando Famílias**, v. 17, n. 1, p. 63–76, 2013. Acessado em: 14 out. 2025.

11 Anexos

Figura 7. Diploma do Bacharelado



Figura 8. Diploma de Técnico

