BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

SISTEMA PARA GESTÃO DE VEÍCULOS APREENDIDOS

SILVESTRE PEREIRA DE OLIVEIRA NETTO

Rio Verde, GO

2026

INSTITUTO FEDERAL GOIANO - CAMPUS RIO VERDE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO

SISTEMA PARA GESTÃO DE VEÍCULOS APREENDIDOS

SILVESTRE PEREIRA DE OLIVEIRA NETTO

Trabalho de Conclusão de Curso apresentado ao Instituto Federal Goiano - Campus Rio Verde, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Orientador: Prof. Adriano Soares de Oliveira Bailao

Rio Verde, GO

Março, 2026

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO

PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO

Com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano a disponibilizar gratuitamente o documento em formato digital no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

IDENTIFICAÇÃO DA PRODUÇÃO TÉCNICO-CIENTÍFICA

Tese (doutorado)	Artigo científico
Dissertação (mestrado)	Capítulo de livro
Monografia (especialização)	Livro
■ TCC (graduação)	Trabalho apresentado em evento

Produto técnico e educacional - Tipo:

Nome completo do autor: SILVESTRE PEREIRA DE OLIVEIRA NETTO

Matrícula: 2022102201940220

Título do trabalho: SISTEMA PARA GESTÃO DE VEÍCULOS APREENDIDOS

RESTRIÇÕES DE ACESSO AO DOCUMENTO

Documento confidencial: ■ Não Sim, justifique:

Informe a data que poderá ser disponibilizado no RIIF Goiano: 20 / 04 / 2024

O documento está sujeito a registro de patente? Sim ■ Não

O documento pode vir a ser publicado como livro? Sim ■ Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O(a) referido(a) autor(a) declara:

• Que o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;

• Que obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autoria, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;

• Que cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Rio Verde - GO
Local

18 / 04 / 2024
Data

_____
Assinatura do autor e/ou detentor dos direitos autorais

Ciente e de acordo:

_____
Assinatura do(a) orientador(a)

Regulamento de Trabalho de Conclusão de Curso (TCC) – IF Goiano - Campus Rio Verde

ANEXO V - ATA DE DEFESA DE TRABALHO DE CURSO

Aos 24 dias do mês de março de dois mil e vinte e seis, às 20h30, reuniu-se a Banca Examinadora composta por: Prof. Dr. Adriano Soares de Oliveira Bailão (orientador), Prof. Me. Caíke da Rocha Damke e Profa. Me. Andrea Barboza Proto Sardi, para examinar o Trabalho de Conclusão de Curso (TCC) intitulado “SISTEMA PARA GESTÃO DE VEÍCULOS APREENDIDOS”, de SILVESTRE PEREIRA DE OLIVEIRA NETTO, estudante do curso de Ciência da Computação do IF Goiano – Campus Rio Verde, sob Matrícula nº 2022102201940220.

A palavra foi concedida ao estudante para a apresentação oral do TC, em seguida houve arguição do candidato pelos membros da Banca Examinadora. Após tal etapa, a Banca Examinadora decidiu pela APROVAÇÃO do estudante.

Ao final da sessão pública de defesa foi lavrada a presente ata, que, após apresentação da versão corrigida do TC, foi assinada pelos membros da Banca Examinadora.

Rio Verde, 24 de março de 2026.

(Assinado eletronicamente)

Adriano Soares de Oliveira Bailão

Orientador(a)

(Assinado eletronicamente)

Caike da Rocha Damke

Membro da Banca Examinadora

(Assinado eletronicamente)

Andrea Barboza Proto Sardi

Membro da Banca Examinadora

Documento assinado eletronicamente por:

- **Adriano Soares de Oliveira Bailao**, **PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 24/03/2026 21:13:31.
- **Caike da Rocha Damke**, **PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 24/03/2026 21:35:11.
- **Andrea Barboza Proto Sardi**, **PROFESSOR ENS BASICO TECN TECNOLOGICO** , em 24/03/2026 21:35:24.

Este documento foi emitido pelo SUAP em 23/03/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifgoiano.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código Verificador: 803252
Código de Autenticação: 99a057baf4

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema Integrado de Bibliotecas do IF Goiano - SIBi

Pereira de Oliveira Netto, Silvestre
N476 Sistema para Gestão de Veículos Apreendidos / Silvestre Pereira de Oliveira Netto. Rio Verde 2026.

53f. il.

Orientador: Prof. Dr. Adriano Soares de Oliveira Bailao.
Monografia (Bacharel) - Instituto Federal Goiano, curso de 0219201 - Bacharelado em Ciência da Computação - Integral - Rio Verde (Campus Rio Verde).
1. Sistemas de Informação. 2. Gestão de Veículos Apreendidos. 3. Polícia Civil. 4. Django. 5. Python. I. Título.

Dedico este trabalho à minha família, que esteve ao meu lado em cada etapa desta jornada, oferecendo suporte incondicional, paciência e encorajamento nos momentos mais desafiadores. Sem a presença e o apoio de cada um de vocês, este caminho não teria sido possível.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por me conceder discernimento, força e perseverança ao longo de toda esta trajetória acadêmica.

À minha família, pelo amor, pelo exemplo e pelo apoio irrestrito em todos os momentos — especialmente nos mais difíceis.

Aos servidores e delegados da 8ª Delegacia Regional de Polícia (8ª DRP) da Polícia Civil do Estado de Goiás, pela receptividade, pela disponibilidade em participar das entrevistas e pelo feedback precioso durante o desenvolvimento e a validação do sistema. A contribuição de vocês foi fundamental para que este trabalho refletisse a realidade operacional da delegacia.

Aos professores do Instituto Federal Goiano — *Campus* Rio Verde, em especial ao meu orientador, pela condução atenciosa deste trabalho, pelos ensinamentos compartilhados ao longo do curso e pela disponibilidade e dedicação com que orientou cada etapa deste projeto.

A todos os docentes do Curso de Bacharelado em Ciência da Computação que contribuíram com minha formação acadêmica e profissional, transmitindo conhecimento, rigor científico e paixão pela área.

Aos colegas de curso, pela parceria, pelas trocas de experiência e pelos momentos de aprendizado conjunto.

Meu sincero reconhecimento a todos que, direta ou indiretamente, contribuíram para a conclusão deste trabalho.

RESUMO

Silvestre, NETTO. **Sistema para Gestão de Veículos Apreendidos.** Março, 2026. 38 f. Monografia – (Curso de Bacharel em Ciência da Computação), Instituto Federal Goiano - Campus Rio Verde. Rio Verde, GO.

A gestão de veículos apreendidos pela Polícia Civil é uma atividade administrativa estratégica que exige precisão, rastreabilidade e segurança no tratamento das informações. Contudo, em muitas unidades policiais, esse controle ainda é realizado de forma manual, por meio de planilhas eletrônicas e registros físicos dispersos, o que gera inconsistências, retrabalho e dificuldades no acesso ágil aos dados. Este trabalho apresenta o desenvolvimento de um sistema web para a gestão de veículos apreendidos, implantado na 8ª Delegacia Regional de Polícia (8ª DRP) da Polícia Civil do Estado de Goiás (PCGO). O sistema foi desenvolvido com a linguagem de programação Python e o *framework* Django, adotando o padrão arquitetural *Model-View-Template* (MVT), com banco de dados PostgreSQL, interface responsiva construída com Bootstrap 5 e geração de documentos em formato PDF por meio da biblioteca ReportLab. A metodologia empregada foi a pesquisa aplicada, de abordagem qualitativa, com coleta de requisitos por meio de entrevistas informais realizadas diretamente com os servidores da unidade policial. O sistema contempla o cadastro completo de veículos, proprietários e procedimentos policiais, além de rastreamento do ciclo de vida de cada bem apreendido — desde o registro da ocorrência até a devolução, encaminhamento para leilão ou descarte. Outras funcionalidades implementadas incluem gestão de imagens com compressão automática, geração de relatórios filtrados, *dashboard* com indicadores em tempo real, controle de acesso por perfil de usuário e trilha de auditoria para todas as operações. A avaliação realizada com os servidores da 8ª DRP demonstrou que a ferramenta reduziu significativamente o tempo necessário para localizar registros, eliminou inconsistências decorrentes dos controles manuais e foi considerada de fácil compreensão e utilização pelos próprios agentes. Os resultados indicam que a solução cumpriu os objetivos propostos, contribuindo para a modernização da gestão de bens apreendidos e podendo servir de modelo para implantação em outras unidades da Polícia Civil. O software foi registrado junto ao INPI em dois módulos — backend (processo nº BR512025006985-5) e frontend (processo nº BR512025006723-2) —, com titularidade do IF Goiano, conferindo proteção formal de propriedade intelectual ao produto desenvolvido.

Palavras-chave: Sistemas de Informação. Gestão de Veículos Apreendidos. Polícia Civil. Django. Python. PostgreSQL. Desenvolvimento Web. Segurança da Informação. Registro de Software.

ABSTRACT

Silvestre, NETTO. Sistema de Gestão. Março, 2026. 38 f. Trabalho de Conclusão de Curso – Bacharel em Ciência da Computação, Instituto Federal Goiano - Campus Rio Verde. Rio Verde, GO, Março, 2026.

The management of seized vehicles by the Civil Police is a strategic administrative activity that demands precision, traceability, and security in the handling of information. However, in many police units, this control is still performed manually through scattered spreadsheets and physical records, which leads to inconsistencies, rework, and difficulties in quickly accessing data. This work presents the development of a web-based system for managing seized vehicles, implemented at the 8th Regional Police Station (8th DRP) of the Civil Police of the State of Goiás (PCGO), Brazil. The system was built using the Python programming language and the Django framework, following the Model-View-Template (MVT) architectural pattern, with a PostgreSQL database, a responsive interface built with Bootstrap 5, and PDF document generation through the ReportLab library. The methodology adopted was applied research with a qualitative approach, collecting requirements through informal interviews conducted directly with the police unit's staff. The system encompasses the complete registration of vehicles, owners, and police procedures, as well as full lifecycle tracking of each seized asset — from the initial record through return to the owner, forwarding to auction, or disposal. Additional features include image management with automatic compression, filtered report generation, a real-time dashboard with key indicators, role-based access control, and an audit trail for all operations. The evaluation conducted with the staff of the 8th DRP demonstrated that the tool significantly reduced the time required to locate records, eliminated inconsistencies caused by manual controls, and was considered intuitive and easy to use by the officers themselves. The results indicate that the solution met its proposed objectives, contributing to the modernization of seized asset management and serving as a model for implementation in other Civil Police units. The software was formally registered with the Brazilian National Institute of Industrial Property (INPI) in two modules — backend (process no. BR512025006985-5) and frontend (process no. BR512025006723-2) —, with the IF Goiano as the holder, granting 50-year intellectual property protection to the developed product.

Keywords: Information Systems. Seized Vehicle Management. Civil Police. Django. Python. PostgreSQL. Web Development. Information Security. Software Registration.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

8ª DRP	8ª Delegacia Regional de Polícia Civil de Goiás
ABNT	Associação Brasileira de Normas Técnicas
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	*Application Programming Interface* (Interface de Programação de Aplicações)
CNH	Carteira Nacional de Habilitação
CPF	Cadastro de Pessoas Físicas
CSRF	*Cross-Site Request Forgery* (Falsificação de Requisição entre Sites)
CSS	*Cascading Style Sheets* (Folhas de Estilo em Cascata)
DER	Diagrama Entidade-Relacionamento
DETRAN	Departamento Estadual de Trânsito
DRY	*Don't Repeat Yourself* (Não Se Repita)
HTML	*HyperText Markup Language* (Linguagem de Marcação de Hipertexto)
IF Goiano	Instituto Federal de Educação, Ciência e Tecnologia Goiano
MVC	*Model-View-Controller* (Modelo-Visão-Controlador)
MVT	*Model-View-Template* (Modelo-Visão-Template)
ORM	*Object-Relational Mapping* (Mapeamento Objeto-Relacional)
OWASP	*Open Worldwide Application Security Project*
PCGO	Polícia Civil do Estado de Goiás
PDF	*Portable Document Format* (Formato Portátil de Documento)
RAI	Registro de Atendimento Integrado
RENAVAM	Registro Nacional de Veículos Automotores
REST	*Representational State Transfer* (Transferência de Estado Representacional)
RF	Requisito Funcional
RNF	Requisito Não Funcional
SGBD	Sistema Gerenciador de Banco de Dados
SQL	*Structured Query Language* (Linguagem de Consulta Estruturada)

SSP/GO	Secretaria de Estado da Segurança Pública de Goiás
TCO	Termo Circunstanciado de Ocorrência
UML	*Unified Modeling Language* (Linguagem de Modelagem Unificada)
XSS	*Cross-Site Scripting* (Injeção de Scripts entre Sites)

SUMÁRIO

1 INTRODUÇÃO

A gestão de veículos apreendidos pela Polícia Civil enfrenta desafios consideráveis, especialmente no que se refere ao controle, rastreamento e documentação desses bens. Com o aumento das operações policiais e o número crescente de apreensões, a gestão manual e o uso de planilhas dispersas tornam-se progressivamente ineficazes, levando a atrasos, erros humanos e dificuldades no acesso às informações necessárias. Esses problemas impactam diretamente a eficiência institucional, comprometendo a rastreabilidade dos veículos apreendidos e o acompanhamento de seu destino final, seja para devolução ao proprietário, encaminhamento para leilão ou descarte.

Atualmente, o processo de controle na 8ª Delegacia Regional de Polícia (8ª DRP) da Polícia Civil do Estado de Goiás (PCGO) é realizado por meio de registros físicos e planilhas eletrônicas, nas quais são inseridas manualmente informações como placa, dados do proprietário e situação do veículo. Embora esse método tenha sido funcional em um contexto de menor volume de apreensões, ele não atende mais às exigências de eficiência e segurança da informação que a realidade atual impõe. A falta de integração entre os dados dispersos e a dificuldade de acesso simultâneo às informações comprometem a agilidade nas operações, tornando o processo mais suscetível a falhas e inconsistências.

Em levantamento realizado na 8ª DRP, constatou-se que a unidade recebe, em média, dezenas de veículos apreendidos por mês. O controle manual desses registros resultava em atrasos significativos: localizar um processo ou gerar um relatório podia levar de 20 a 30 minutos, além de exigir a conferência de várias planilhas separadas. Esses dados evidenciam a necessidade de informatizar o processo. A digitalização permite reduzir o tempo de busca e aumentar a confiabilidade das informações, contribuindo para a celeridade das investigações e a eficiência da gestão de bens apreendidos.

Diante dessa realidade, este trabalho propõe o desenvolvimento de um Sistema de Gestão de Veículos Apreendidos para a Polícia Civil do Estado de Goiás. Utilizando o *framework* Django e a linguagem Python, a solução centraliza em uma única base de dados o cadastro completo de veículos, proprietários e procedimentos policiais, rastreando cada bem desde a apreensão até sua devolução ou destinação final. Funcionalidades como atualização em tempo real do status de cada veículo, geração automática de relatórios em PDF e uma interface intuitiva eliminam o uso de planilhas e documentos físicos, garantindo maior precisão, agilidade e segurança na gestão dos bens apreendidos.

Além disso, o sistema foi desenvolvido com foco em segurança da informação, com controles de acesso baseados em perfis de usuário, restrições a dados sensíveis e registro de trilha de auditoria para todas as operações realizadas. Dessa forma, garante-se que os dados sejam acessados de forma organizada e segura, sem comprometer sua integridade.

1.1 Objetivo Geral

Desenvolver um sistema web para a gestão centralizada e rastreável de veículos apreendidos pela 8ª Delegacia Regional de Polícia (8ª DRP) da Polícia Civil do Estado de Goiás, substituindo os controles manuais e em planilhas por uma plataforma informatizada que garanta maior eficiência, segurança e rastreabilidade em todas as etapas do processo de custódia dos bens.

1.1.1 Objetivos Específicos

- Levantar e analisar os requisitos funcionais e não funcionais do sistema por meio de entrevistas com os servidores e agentes responsáveis pelo controle de veículos apreendidos na 8ª DRP;
- Modelar a arquitetura do sistema com base no padrão *Model-View-Template* (MVT) do *framework* Django, elaborando diagramas de casos de uso, diagrama de classes e modelo entidade-relacionamento do banco de dados;
- Implementar as funcionalidades centrais do sistema, incluindo cadastro de veículos, proprietários e procedimentos policiais, controle de status e ciclo de vida dos bens, gestão de imagens e geração de documentos em PDF;
- Desenvolver mecanismos de segurança da informação, incluindo autenticação de usuários, controle de acesso por perfil e trilha de auditoria das operações realizadas no sistema;
- Validar o sistema junto aos usuários finais da 8ª DRP, avaliando a usabilidade, a aderência aos requisitos definidos e a eficácia da ferramenta como substituta dos controles manuais;
- Implantar o sistema em ambiente de produção e treinar os servidores da unidade para utilização adequada da plataforma.

1.2 Justificativa

A necessidade de modernizar o controle de veículos apreendidos pela Polícia Civil do Estado de Goiás é motivada pelas limitações evidentes dos processos manuais vigentes. O uso de planilhas eletrônicas desconectadas e de registros físicos resulta em inconsistências nos dados, dificuldade de acesso simultâneo às informações e ausência de rastreabilidade das operações realizadas. Em um ambiente institucional em que a precisão e a agilidade das informações impactam diretamente os procedimentos policiais e jurídicos, essas deficiências representam riscos operacionais concretos.

A implantação de um sistema informatizado centralizado permite otimizar o processo de registro e consulta, eliminar retrabalho e reduzir significativamente o tempo demandado para localizar informações sobre veículos apreendidos. Além disso, a adoção de controles de acesso e trilha de auditoria confere maior segurança e *accountability* às operações, atributos essenciais em contextos de gestão pública.

Do ponto de vista acadêmico, o projeto contribui com a aplicação prática de conceitos de Engenharia de Software, desenvolvimento web e segurança da informação em um caso de uso real e socialmente relevante, demonstrando que soluções tecnológicas desenvolvidas com metodologias adequadas e participação ativa dos usuários podem transformar processos administrativos e melhorar a qualidade dos serviços públicos.

2 REVISÃO DE LITERATURA

Este capítulo apresenta uma síntese da literatura relacionada ao tema deste trabalho, abrangendo os fundamentos teóricos de Engenharia de Software, desenvolvimento web com Django e Python, gerenciamento de banco de dados relacionais, metodologias ágeis de desenvolvimento e trabalhos correlatos que tratam de sistemas de gestão de veículos apreendidos ou de frotas em geral. A revisão busca situar este trabalho no estado da arte e justificar as escolhas metodológicas e tecnológicas adotadas.

2.1 Engenharia de Software e Processos de Desenvolvimento

A Engenharia de Software, conforme definida por Sommerville (2011), compreende um conjunto de disciplinas voltadas para o desenvolvimento sistemático de softwares de qualidade, englobando processos, métodos, ferramentas e práticas de gestão de projetos. Dentre os princípios fundamentais destacam-se a modularidade, a manutenibilidade e a rastreabilidade dos requisitos ao longo de todo o ciclo de vida do sistema.

Pressman e Maxim (2016) ampliam essa visão ao classificar os processos de software em modelos prescritivos (como o modelo cascata e o espiral) e modelos ágeis, ressaltando que a escolha do processo deve considerar o porte do projeto, a estabilidade dos requisitos e o grau de envolvimento dos usuários. Para projetos institucionais com requisitos evolutivos e usuários acessíveis para validação contínua, os autores recomendam abordagens iterativas e incrementais, o que embasou a adoção de metodologia incremental neste trabalho.

2.2 Metodologias Ágeis e Desenvolvimento Incremental

O Desenvolvimento Ágil, conceituado por Sommerville (2011) e popularizado pelo Manifesto Ágil (BECK et al., 2001), propõe a entrega de funcionalidades em ciclos curtos, com validação frequente junto ao cliente. Essa abordagem é adequada para contextos em que os requisitos não estão completamente definidos no início do projeto, permitindo que ajustes sejam realizados a cada iteração.

Schwaber e Sutherland (2020) descrevem o *Scrum* como um dos *frameworks* ágeis mais adotados, estruturando o desenvolvimento em *sprints* com papéis, eventos e artefatos bem definidos. Embora o presente trabalho não tenha adotado o *Scrum* formalmente, o modelo incremental aplicado compartilha de seus princípios centrais: entregas parciais funcionais, validação com os usuários reais e adaptação contínua dos requisitos.

2.3 Desenvolvimento Web com Python e Django

Python é uma linguagem de programação de alto nível, interpretada e com sintaxe de alta legibilidade. Lutz (2013) destaca sua versatilidade e a riqueza de seu ecossistema de bibliotecas como fatores que a tornam adequada para projetos que demandam desenvolvimento ágil e código de fácil manutenção.

O Django é um *framework* web de alto nível para Python que incentiva o desenvolvimento rápido e o *design* limpo e pragmático. Segundo a documentação oficial (Django Software Foundation, 2024), ele segue o princípio *Don't Repeat Yourself* (DRY) e

oferece, de forma integrada, um sistema de autenticação, um ORM (Mapeamento Objeto-Relacional) para comunicação com o banco de dados e proteções nativas contra vulnerabilidades como injeção de SQL, *Cross-Site Scripting* (XSS) e falsificação de requisição entre sites (CSRF), além de um painel administrativo configurável.

2.3.1 Padrão Arquitetural MVT e sua relação com o MVC

O padrão *Model-View-Controller* (MVC), descrito por Gamma et al. (1995) em *Design Patterns: Elements of Reusable Object-Oriented Software*, é uma das arquiteturas mais difundidas no desenvolvimento de software. Nesse padrão, o **Model** representa os dados e as regras de negócio; a **View** é responsável pela apresentação das informações ao usuário; e o **Controller** intermedia as interações, recebendo as requisições do usuário, manipulando o *Model* e decidindo qual *View* exibir.

O Django adota uma variação desse padrão denominada *Model-View-Template* (MVT). A diferença fundamental está na nomenclatura e na responsabilidade dos componentes: o que no MVC é chamado de **Controller** corresponde, no Django, às **Views** — funções ou classes Python que processam as requisições HTTP e retornam respostas. Já o que o MVC denomina **View** (camada de apresentação) é representado, no Django, pelos **Templates** — arquivos HTML com marcações especiais que permitem a renderização dinâmica de dados. O **Model** mantém a mesma função nos dois padrões: definir a estrutura dos dados e a lógica de persistência. A Tabela 1 sintetiza essa correspondência.

Tabela 1 – Comparativo entre os padrões MVC e MVT (Django)

Responsabilidade	**MVC**	**MVT (Django)**
Dados e regras de negócio	Model	Model
Lógica de controle / processamento	Controller	View
Apresentação ao usuário	View	Template

Fonte: Adaptado de Gamma et al. (1995) e Django Software Foundation (2024).

Portanto, embora os nomes divirjam, ambos os padrões compartilham o mesmo princípio fundamental: a separação de responsabilidades (*Separation of Concerns*), que facilita a manutenção, os testes e a evolução independente de cada camada do sistema.

2.4 Banco de Dados Relacional e PostgreSQL

O PostgreSQL é um Sistema Gerenciador de Banco de Dados (SGBD) objeto-relacional de código aberto, robusto e amplamente adotado em aplicações que exigem alta confiabilidade e integridade dos dados. Date (2004) destaca que os bancos de dados relacionais garantem integridade referencial por meio do uso de chaves primárias e estrangeiras, consistência das transações com suporte às propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) e uma linguagem padronizada de consulta e manipulação de dados — a SQL (*Structured Query Language*).

Sua adoção neste trabalho foi motivada pela maturidade do sistema, pela compatibilidade nativa com o *framework* Django e pela capacidade de lidar com consultas complexas e grandes volumes de dados de forma eficiente.

2.5 Segurança da Informação em Sistemas Web

A segurança da informação é um requisito não funcional crítico para sistemas que gerenciam dados sensíveis, especialmente em contextos de gestão pública. O *Open Web Application Security Project* (OWASP, 2021) publica periodicamente o *OWASP Top 10*, um ranking das vulnerabilidades mais críticas em aplicações web, que inclui falhas como injeção de SQL, autenticação inadequada, exposição de dados sensíveis e controle de acesso falho.

O Django responde a várias dessas ameaças de forma nativa, como discutido por Forcier, Bissex e Chun (2008), que ressaltam que o *framework* adota boas práticas de segurança por padrão, cabendo ao desenvolvedor configurar corretamente os mecanismos disponíveis. As decisões de segurança implementadas neste sistema — autenticação, controle de acesso por perfil, validação de arquivos enviados e registro de auditoria — são diretamente embasadas nessas recomendações.

2.6 Trabalhos Correlatos

A literatura acerca de sistemas de controle de veículos apreendidos e de gestão de frotas revela um esforço crescente em substituir processos manuais por soluções informatizadas mais ágeis, seguras e transparentes.

Veríssimo (2013) apresentou uma solução pioneira para o controle de veículos apreendidos na 17ª Companhia Independente de Polícia Militar (CIPM), combinando modelos de processo tradicionais com metodologias iterativas. O autor utilizou a Linguagem de Modelagem Unificada (UML) para modelar os requisitos e realizou entrevistas com usuários na etapa de elicitação. A solução implementou auditoria no banco de dados, assegurando rastreabilidade das operações.

Silva e Barbosa (2018) desenvolveram um sistema web para a Delegacia de Repressão a Roubos e Furtos de Veículos da Polícia Civil do Pará, adotando práticas de *Scrum* e ferramentas colaborativas. A solução, construída com o *framework* Zend, HTML5, jQuery e Bootstrap, exemplifica como métodos ágeis podem ser combinados com tecnologias consolidadas para entregar sistemas robustos.

Neto (2022) desenvolveu um sistema de gestão de frota para a Escola Agrícola de Jundiaí, utilizando Python com Django e contêineres Docker, com validações contínuas junto aos usuários. O autor concluiu que a informatização melhora a comunicação interna e reduz o uso de papel, conclusão alinhada com os resultados observados no presente trabalho.

Por fim, Silva (2022) abordou o gerenciamento de fretes em uma transportadora utilizando Python e Django com padrão MVT, MySQL como banco de dados e Bootstrap para a interface, criando um sistema robusto para cadastro de motoristas, fretes, abastecimentos e manutenções.

Dessa síntese emerge um conjunto de boas práticas consolidado: elicitação de requisitos com participação efetiva dos usuários; modelagem formal em UML; adoção de metodologias ágeis combinadas com prototipação; e integração de tecnologias web modernas com mecanismos de auditoria e segurança. O presente trabalho foi desenvolvido tendo essas diretrizes como referência.

3 MATERIAIS E MÉTODOS

A materialização da solução proposta para a gestão de veículos na 8ª DRP fundamentou-se em práticas consolidadas de Engenharia de Software, priorizando a estabilidade, a segurança da informação e a escalabilidade. A arquitetura do sistema foi estruturada para suportar o fluxo completo de apreensão e custódia, integrando tecnologias de desenvolvimento web modernas a uma metodologia de trabalho iterativa, capaz de alinhar os requisitos técnicos às necessidades operacionais da Polícia Civil.

3.1 Metodologia de Desenvolvimento

3.1.1 Caracterização da Pesquisa

O presente trabalho caracteriza-se como uma pesquisa aplicada, de abordagem qualitativa, com natureza exploratória e descritiva, desenvolvida por meio de um estudo de caso institucional.

A pesquisa aplicada foi adotada por ter como finalidade a resolução de um problema concreto e atual, identificado no contexto da gestão de veículos apreendidos no âmbito da Polícia Civil, buscando não apenas compreender a realidade existente, mas propor e implementar uma solução tecnológica funcional.

A abordagem qualitativa foi empregada por se basear na análise de processos, fluxos de trabalho e percepções dos usuários diretamente envolvidos na atividade operacional. Para isso, realizou-se um levantamento das práticas adotadas na atualidade, com o objetivo de compreender como ocorria o controle de veículos e peças apreendidas, identificar limitações, gargalos operacionais e riscos associados ao uso de controles manuais e planilhas.

A coleta de dados ocorreu por meio de entrevistas informais e consultas diretas aos servidores e funcionários responsáveis pelo registro, controle e armazenamento dos veículos, permitindo identificar dificuldades recorrentes, necessidades não atendidas e sugestões de melhorias. Essas informações foram fundamentais para a definição dos requisitos do sistema, priorização de funcionalidades e modelagem das regras de negócio.

O estudo de caso institucional permitiu observar o ambiente real de aplicação do sistema, garantindo que o desenvolvimento estivesse alinhado às rotinas administrativas existentes e às demandas operacionais da unidade policial, resultando em uma solução aderente à realidade, validada pelos próprios usuários ao longo do processo de desenvolvimento.

3.1.2 Processo de Desenvolvimento

O desenvolvimento do sistema adotou o modelo incremental de desenvolvimento de software, no qual as funcionalidades foram construídas, integradas e validadas em ciclos sucessivos. Não foi aplicado formalmente nenhum *framework* ágil com cerimônias predefinidas, como o Scrum; optou-se, em seu lugar, por um processo pragmático e orientado à entrega. A organização das etapas seguiu uma progressão lógica e predominantemente sequencial — levantamento de requisitos, modelagem, design de interface, implementação e testes —, pois cada fase fornecia insumos essenciais para a seguinte. Contudo, a fase de implementação e testes foi conduzida de forma incremental: os módulos foram desenvolvidos, entregues e validados progressivamente, sem aguardar a conclusão de todos os

demais para iniciar a validação. Assim, houve entregas contínuas de funcionalidades ao longo do desenvolvimento, com retorno imediato dos servidores da 8ª DRP a cada novo módulo disponibilizado. O projeto foi estruturado nas seguintes etapas:

1. **Levantamento de Requisitos:** A primeira etapa envolveu a coleta dos requisitos necessários para o sistema, realizada por meio de entrevistas informais com os usuários finais (servidores da 8ª Delegacia Regional de Polícia Civil de Goiás) e análise dos processos manuais existentes, como o uso de planilhas para controle de veículos apreendidos. A definição de requisitos funcionais e não funcionais foi essencial para a modelagem do sistema e a escolha das ferramentas de desenvolvimento. Esta etapa foi conduzida de forma sequencial e anterior às demais, pois os requisitos levantados orientaram as decisões de modelagem e arquitetura.
2. **Modelagem do Sistema:** Após o levantamento dos requisitos, utilizou-se a abordagem Model-View-Template (MVT) para estruturar o desenvolvimento. A modelagem incluiu a criação de diagramas de casos de uso, diagramas de classes e fluxogramas de atividades, que descrevem as interações entre os usuários e o sistema e serviram de referência durante a implementação.
3. **Prototipação e Design:** A fase de design de interface foi realizada antes da implementação das telas, com o objetivo de validar a estrutura de navegação e o fluxo das principais funcionalidades. Foram elaborados esboços e protótipos de baixa fidelidade (wireframes) das telas principais, incluindo login, cadastro de veículos, consulta de documentos e geração de relatórios em PDF, utilizando como referência visual ferramentas de edição de imagem e esboço manual. Esses protótipos foram apresentados e discutidos com os servidores da 8ª DRP antes do início da implementação, permitindo ajustes no fluxo de navegação e na organização das informações. O design final foi implementado com o *framework* Bootstrap, que garantiu responsividade e consistência visual em todas as telas.
4. **Implementação:** O desenvolvimento do sistema foi realizado de forma incremental, módulo a módulo. A escolha do *framework* Django — abordada em detalhes na Seção 3.2 — determinou a adoção da linguagem Python, uma vez que o Django é nativamente implementado nessa linguagem. A interação com o banco de dados foi realizada por meio do ORM do Django em conjunto com o PostgreSQL. Cada módulo concluído era disponibilizado para uso e avaliação pelos servidores, possibilitando retorno contínuo e ajustes durante o próprio ciclo de desenvolvimento.
5. **Testes:** Os testes do sistema foram conduzidos pelos próprios servidores da 8ª DRP ao longo do desenvolvimento incremental, garantindo validação contínua das funcionalidades em ambiente real de uso. Os usuários finais — agentes de polícia e servidores administrativos — participaram ativamente das sessões de teste, avaliando cada módulo implementado e fornecendo retorno imediato sobre adequação às rotinas operacionais da delegacia. A metodologia de avaliação, os critérios de aceitação e a descrição detalhada das sessões de teste encontram-se no Capítulo 4.

3.1.3 Tipos de Testes Realizados

Os testes conduzidos ao longo do desenvolvimento foram organizados em três categorias principais, cada uma com foco e critérios de avaliação distintos:

Testes funcionais: Verificaram se cada funcionalidade do sistema opera conforme os requisitos definidos. Foram estruturados em 10 cenários de teste (CT01–CT10), cobrindo autenticação, cadastro, validação de formulários, *upload* de imagens, consulta e filtragem de registros, geração de relatórios em PDF, controle de acesso por perfil, exclu-

são auditada e visualização do painel de indicadores (*dashboard*). Todos os cenários foram executados diretamente pelos servidores da 8ª DRP em ambiente real, sem simulações ou dados fictícios.

Testes de desempenho: Avaliaram a capacidade de resposta do sistema sob condições de uso real. O critério adotado foi o tempo de resposta inferior a 3 segundos para consultas e listagens, conforme estabelecido no RNF02. Durante as sessões de teste com os servidores, nenhuma operação registrou tempo de resposta superior ao limite estipulado, confirmando a adequação das estratégias de paginação e otimização de consultas implementadas.

Testes de usabilidade: Avaliaram a facilidade de uso do sistema por parte dos usuários finais, sem treinamento formal prévio. Os servidores realizaram tarefas reais do cotidiano operacional e avaliaram a compreensibilidade da interface, a clareza das mensagens de erro e de confirmação, e a capacidade de concluir as tarefas de forma autônoma. Os resultados e critérios de aceitação adotados estão detalhados no Capítulo 4.

Os cenários de teste foram organizados em dois grupos. A Tabela 2 apresenta os cenários relativos a autenticação, cadastro e validação de dados; a Tabela 3 apresenta os cenários de consulta, geração de documentos e controle de acesso.

Tabela 2 – Cenários de teste — Autenticação, Cadastro e Validação (CT01–CT05)

ID	**Funcionalidade**	**Cenário de Teste**	**Resultado**
CT01	Autenticação	Login com credenciais válidas; redirecionamento ao painel	Aprovado
CT02	Autenticação	Login com senha incorreta; acesso negado	Aprovado
CT03	Cadastro de veículo	Registro completo com placa, RENAVAM, chassi, proprietário e procedimento	Aprovado
CT04	Validação de campos	Formulário enviado sem campos obrigatórios; mensagem de erro exibida	Aprovado
CT05	Upload de imagens	Envio de imagem JPG até 5 MB; miniatura gerada automaticamente	Aprovado

Fonte: Elaboração própria, com base nos testes realizados com os servidores da 8ª DRP (2026).

3.2 Ferramentas e Tecnologias Utilizadas

As ferramentas e tecnologias adotadas no desenvolvimento do sistema foram selecionadas considerando critérios como desempenho, escalabilidade, segurança da informação e aderência ao contexto institucional da Polícia Civil. A opção por soluções consolidadas e amplamente utilizadas no mercado contribui para a manutenção futura do sistema e para sua possível expansão.

3.2.1 Backend e Camada de Negócio

Para o desenvolvimento do *backend* da aplicação web, foi escolhido o *framework* Django (Figura 1), que estabeleceu a adoção da linguagem Python (Figura 2) como consequência direta: o Django é desenvolvido integralmente em Python e exige essa linguagem

Tabela 3 – Cenários de teste — Consulta, Relatórios e Controle de Acesso (CT06–CT10)

ID	**Funcionalidade**	**Cenário de Teste**	**Resultado**
CT06	Busca e filtro	Filtragem por placa, status e unidade; resultado correspondente exibido	Aprovado
CT07	Geração de PDF	Relatório com filtros aplicados; *download* do arquivo disponibilizado	Aprovado
CT08	Controle de acesso	Operador tenta excluir veículo; operação bloqueada pelo sistema	Aprovado
CT09	Exclusão com auditoria	Administrador exclui registro informando motivo; log de auditoria gravado	Aprovado
CT10	Dashboard	Totais e indicadores do painel exibidos corretamente	Aprovado

Fonte: Elaboração própria, com base nos testes realizados com os servidores da 8ª DRP (2026).

como ambiente de execução. Dessa forma, a escolha tecnológica partiu do *framework* — selecionado por sua maturidade, segurança nativa e amplo suporte a desenvolvimento web — e a linguagem foi adotada em razão dele, e não de forma independente.

O Django adota o padrão arquitetural Model-View-Template (MVT), que estabelece uma separação clara entre três responsabilidades distintas: a camada de dados (*Model*), a lógica de processamento (*View*) e a apresentação visual (*Template*). Essa estrutura facilita a organização do código, reduz o acoplamento entre componentes e favorece a manutenção e a evolução do sistema ao longo do tempo. O *framework* também oferece, de forma nativa, proteção contra as principais vulnerabilidades web — incluindo injeção de SQL, *Cross-Site Scripting* (XSS) e CSRF —, além de sistema de autenticação de usuários e controle de permissões, conforme detalhado na Seção 2.5 da Revisão de Literatura.

Figura 1 – Logotipo do Framework Django

Figura 2 – Logotipo da linguagem Python

O mapeamento objeto-relacional (ORM) do Django atua como uma camada de abstração entre a aplicação e o banco de dados relacional. Por meio dela, as entidades do

domínio são representadas como classes Python (subclasses de `django.db.models.Model`), e as operações de leitura, inserção, atualização e exclusão de dados são realizadas por meio de métodos orientados a objetos, sem a necessidade de escrever instruções SQL diretamente. Essa abordagem elimina uma classe inteira de erros decorrentes de consultas SQL malformadas, facilita a portabilidade entre diferentes sistemas gerenciadores de banco de dados e aumenta a confiabilidade das operações de persistência de dados. No sistema desenvolvido, todas as interações com o PostgreSQL foram realizadas exclusivamente via ORM.

3.2.2 Banco de Dados

Para o armazenamento das informações do sistema, foi adotado o Sistema Gerenciador de Banco de Dados relacional PostgreSQL (Figura 3). A escolha do PostgreSQL deve-se à sua confiabilidade, conformidade com o padrão ACID e capacidade de lidar com consultas complexas, características essenciais para o armazenamento de dados sensíveis relacionados a veículos apreendidos, proprietários, procedimentos policiais e registros administrativos.

A estrutura do banco de dados foi modelada de forma a garantir integridade referencial entre as tabelas, permitindo rastreabilidade das informações e consistência nos registros.

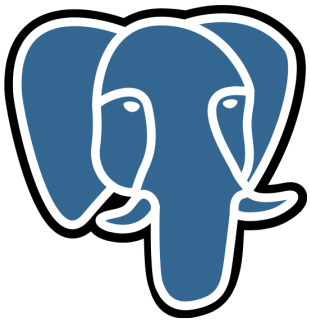

Figura 3 – Logotipo do Banco de dados PostgreSQL

3.2.3 Frontend e Interface do Usuário

A camada de apresentação do sistema foi desenvolvida com HTML5, CSS3 e JavaScript, priorizando uma estrutura semântica adequada e facilidade de navegação. O framework Bootstrap foi utilizado como principal base para o desenvolvimento visual da aplicação, sendo responsável pela padronização dos componentes gráficos, como formulários, botões, tabelas e menus.

O uso do Bootstrap permitiu a criação de uma interface responsiva, adaptável a diferentes tamanhos de tela, atendendo tanto ao uso em computadores quanto em dispositivos móveis utilizados no ambiente institucional. A utilização de JavaScript contribuiu para melhorar a experiência do usuário, permitindo interações dinâmicas e reduzindo a necessidade de recarregamento das páginas.

Para aprimorar a usabilidade e facilitar a identificação visual de funcionalidades, foi utilizado o conjunto de ícones Font Awesome, aplicado em botões e menus do sistema. A Figura 4 apresenta os logotipos das principais tecnologias utilizadas na camada de frontend do sistema.

HTML5 CSS3 JavaScript Bootstrap 5

Figura 4 – Logotipos das tecnologias utilizadas na camada de frontend do sistema

3.2.4 Gerenciamento de Imagens e Documentos

O sistema permite o armazenamento e a visualização de imagens associadas aos veículos apreendidos. Para esse fim, foi utilizada a biblioteca Pillow, responsável pelo processamento e pela otimização das imagens enviadas pelos usuários, contribuindo para a redução do espaço de armazenamento e melhor desempenho da aplicação.

Além disso, o sistema possibilita a geração de documentos em formato PDF, tais como relatórios e registros administrativos. Para essa funcionalidade, foi utilizada a biblioteca ReportLab, amplamente empregada em aplicações Python para a criação programática de documentos PDF. O uso do ReportLab permite a geração de arquivos com formatação controlada, garantindo padronização visual, organização das informações e adequação dos documentos para impressão e arquivamento oficial.

3.2.5 Ambiente de Desenvolvimento e Controle de Versão

O ambiente de desenvolvimento do sistema foi padronizado por meio do uso de contêineres Docker (Figura 5), o que possibilitou a criação de um ambiente isolado e consistente para execução da aplicação. Essa abordagem contribuiu para a redução de problemas de incompatibilidade entre diferentes máquinas e facilitou a replicação do sistema em ambientes distintos, como desenvolvimento, testes e implantação.

Figura 5 – Logotipo do Docker

A implementação do sistema foi realizada utilizando o editor de código Visual Studio Code (VS Code, Figura 6), escolhido por oferecer suporte eficiente à linguagem Python, integração com ferramentas de versionamento e recursos que auxiliam na organização e produtividade durante o desenvolvimento do software.

Figura 6 – Logotipo do Visual Studio Code

O código-fonte do projeto foi versionado utilizando o sistema de controle de versões Git e hospedado no GitHub (Figura 7), ferramentas amplamente adotadas no desenvolvimento de software por permitir o gerenciamento eficiente das alterações realizadas ao longo do ciclo de vida do sistema. O uso do Git possibilitou o acompanhamento do histórico de modificações, a identificação de versões estáveis e a recuperação de estados anteriores do projeto, contribuindo para a organização e confiabilidade do desenvolvimento.

Como plataforma de hospedagem do repositório, foi utilizado o GitHub, que ofereceu suporte ao armazenamento seguro do código-fonte em repositório privado. A utilização do GitHub facilitou o controle das evoluções do sistema, o gerenciamento de versões e a centralização do projeto, além de reforçar a segurança e a rastreabilidade das alterações realizadas.

Logotipo do Git

Logotipo do GitHub

Figura 7 – Ferramentas utilizadas para versionamento e hospedagem do código-fonte

3.3 Requisitos do Sistema

Os requisitos do sistema foram levantados a partir de entrevistas informais com os servidores da 8ª DRP e da análise dos processos manuais previamente adotados. A especificação foi organizada em duas categorias: requisitos funcionais (RF), que descrevem as funcionalidades esperadas, e requisitos não funcionais (RNF), que estabelecem critérios de qualidade, desempenho e segurança.

3.3.1 Requisitos Funcionais

A Tabela 4 apresenta os requisitos funcionais identificados para o sistema, especificando as funcionalidades que devem ser suportadas pela aplicação. O levantamento foi conduzido por meio de entrevistas informais com os servidores da 8ª DRP, observação direta dos processos manuais existentes e análise de documentos internos utilizados na rotina operacional. Cada requisito foi priorizado com base na frequência de uso, no impacto

sobre as atividades diárias dos servidores e na necessidade de substituir controles manuais por funcionalidades informatizadas, garantindo que o sistema atendesse às demandas reais da unidade.

Tabela 4 – Requisitos Funcionais do Sistema

ID	**Descrição**
RF01	O sistema deve permitir o cadastro de veículos apreendidos com dados completos: placa, RENAVAM, chassi, marca, modelo, cor, ano e tipo.
RF02	O sistema deve registrar e associar o proprietário do veículo, incluindo nome completo, CPF, telefone e endereço.
RF03	O sistema deve registrar procedimentos policiais vinculados ao veículo, incluindo número de RAI, TCO, delegacia responsável, data e agente responsável.
RF04	O sistema deve permitir *upload* e visualização de imagens e documentos associados ao veículo apreendido.
RF05	O sistema deve gerar relatórios administrativos em formato PDF, com seleção de filtros como período, status ou unidade responsável.
RF06	O sistema deve permitir busca e filtragem de veículos por placa, status, unidade, tipo e período de apreensão.
RF07	O sistema deve manter histórico de alterações e registro de auditoria para operações sensíveis, como exclusão de registros.
RF08	O sistema deve gerenciar múltiplas unidades (delegacias e pátios) com controle de acesso por unidade.
RF09	O sistema deve fornecer painel de indicadores (*dashboard*) com totais de veículos por status, unidade e período.
RF10	O sistema deve suportar controle de acesso baseado em perfis de usuário (operador e administrador), com restrições para operações críticas.

Fonte: Elaboração própria (2026).

3.3.2 Requisitos Não Funcionais

A Tabela 5 apresenta os requisitos não funcionais do sistema, que estabelecem os critérios de qualidade, desempenho e segurança esperados. Esses requisitos foram definidos a partir das restrições operacionais da unidade, das exigências de segurança da informação aplicáveis a sistemas de uso policial e das expectativas dos usuários quanto à responsividade e à confiabilidade da aplicação. A conformidade com os requisitos não funcionais é essencial para que o sistema possa ser adotado em ambiente real sem comprometer a integridade dos dados ou a experiência dos usuários.

Tabela 5 – Requisitos Não Funcionais do Sistema

ID	**Categoria**	**Descrição e Métrica de Validação**
RNF01	Segurança	Autenticação com expiração de sessão após 30 minutos de inatividade (`SESSION_COOKIE_AGE = 1800`), senhas armazenadas com PBKDF2+SHA-256, proteção automática contra CSRF (middleware), XSS (escape automático de templates) e injeção de SQL (uso exclusivo do ORM). *Métrica:* todos os cenários de segurança dos testes CT01, CT02, CT08 e CT09 aprovados em ambiente real.
RNF02	Desempenho	Consultas e listagens com tempo de resposta inferior a 3 segundos em condições normais de uso, garantido por paginação (`django.core.paginator`) e cache para grandes volumes de dados. *Métrica:* tempo de resposta verificado nas sessões de teste com os servidores da 8ª DRP; nenhum cenário de teste registrou tempo superior ao limite estabelecido.
RNF03	Usabilidade	Interface responsiva e compatível com dispositivos móveis e desktops (Bootstrap 5), garantindo uso intuitivo pelos servidores sem necessidade de treinamento formal extenso. *Métrica:* taxa de conclusão de tarefas de 100% nos 10 cenários de teste (CT01–CT10) avaliados com 6 servidores; todos os usuários atingiram independência operacional em poucas sessões.
RNF04	Disponibilidade	Execução em ambiente de produção via contêineres Docker, com implantação consistente e recuperável. *Métrica:* sistema disponível continuamente durante todo o período de uso pelos servidores da 8ª DRP, sem interrupções registradas.
RNF05	Manutenção	Código seguindo o padrão MVT do Django, com separação de responsabilidades entre camadas e princípio DRY (*Don't Repeat Yourself*), garantindo que cada regra de negócio seja implementada em um único ponto do sistema. *Métrica:* ausência de duplicação de lógica de negócio entre *views* e *templates*, verificada por revisão da estrutura de código.

Fonte: Elaboração própria (2026).

3.3.3 Uso de Ferramentas de Inteligência Artificial

Durante o desenvolvimento deste trabalho, ferramentas de inteligência artificial generativa foram utilizadas como apoio complementar em diferentes etapas do processo. Assistentes baseados em modelos de linguagem de grande escala (*Large Language Models* — LLMs) auxiliaram na sugestão de trechos de código, na revisão de lógica de implementação, na identificação de possíveis melhorias de estrutura e na escrita técnica de partes do texto. O uso dessas ferramentas foi estritamente complementar à autoria dos desenvolvedores: todas as decisões de projeto, a definição de requisitos, a arquitetura do sistema e o julgamento sobre adequação das soluções foram realizados pelos próprios autores, com base no conhecimento técnico adquirido ao longo da formação acadêmica. As sugestões geradas pelas ferramentas de IA foram sempre revisadas criticamente, adaptadas ao contexto do sistema e ao ambiente institucional da 8ª DRP, e validadas antes da incorporação ao produto final. Assim, o uso de inteligência artificial não substituiu o raciocínio crítico dos autores, mas contribuiu para a produtividade e a qualidade do desenvolvimento.

3.4 Modelagem e Arquitetura do Sistema

A arquitetura do sistema de Gestão de Veículos Apreendidos foi concebida com base em princípios de engenharia de software que priorizam organização, modularidade, manutenção e possibilidade de evolução futura. A aplicação foi desenvolvida em Python, utilizando o framework Django, explorando o padrão arquitetural Model-View-Template (MVT), amplamente adotado em aplicações web desenvolvidas com esse framework, conforme ilustrado na Figura 8.

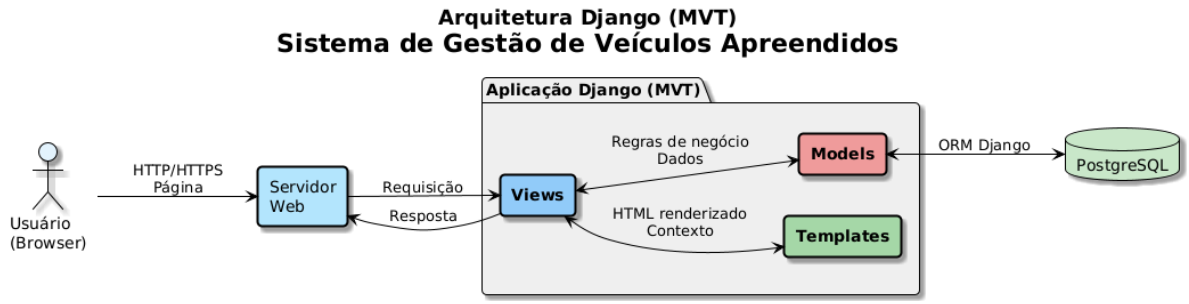

Figura 8 – Arquitetura geral do sistema de gestão de veículos apreendidos

No padrão MVT, os *Models* representam as entidades centrais do domínio do sistema, sendo responsáveis pela definição da estrutura dos dados, regras básicas de negócio e persistência no banco de dados relacional. No sistema desenvolvido, todas as entidades estendem a classe `django.db.models.Model`, e os campos são declarados por meio de tipos específicos do ORM, como `CharField`, `DateField`, `ForeignKey` e `ImageField`. Entre as principais entidades modeladas destacam-se *Veículo*, *Proprietário* e *Procedimento*, que armazenam informações relacionadas às apreensões, à identificação dos proprietários e aos trâmites administrativos associados. Essas entidades concentram a "fonte única da verdade" do sistema, sendo utilizadas tanto pela camada de apresentação quanto por eventuais integrações, conforme representado na Figura 9.

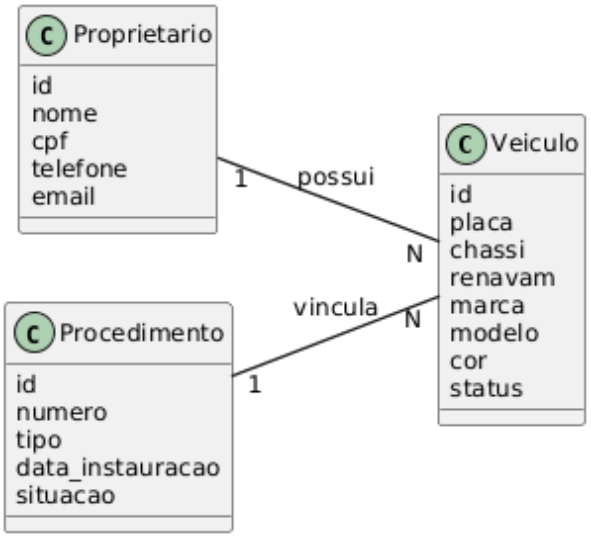

Figura 9 – Diagrama de classes ilustrando as entidades Veículo, Proprietário e Procedimento e seus relacionamentos.

A Figura 10 apresenta o Diagrama Entidade-Relacionamento (DER) completo do banco de dados, detalhando os atributos e os relacionamentos entre todas as entidades modeladas no sistema.

As *Views* desempenham o papel de controladores da aplicação, sendo responsáveis por receber as requisições dos usuários, aplicar regras de autenticação e autorização, realizar filtragens, paginação de dados e preparar o contexto necessário para a renderização das páginas. As *views* do sistema utilizam os módulos `django.contrib.auth` — para autenticação e controle de acesso por meio dos decoradores `login_required` e `permission_required` — e `django.core.paginator`, responsável pela paginação das listagens de veículos e procedimentos. Nessa camada também são implementadas funcionalidades específicas, como a geração de relatórios administrativos em formato PDF via ReportLab e o carregamento de métricas utilizadas nos painéis de acompanhamento (*dashboards*). Os diagramas de casos de uso do Operador e do Administrador (Figuras 11 e 12) detalham as interações previstas para cada perfil de usuário do sistema.

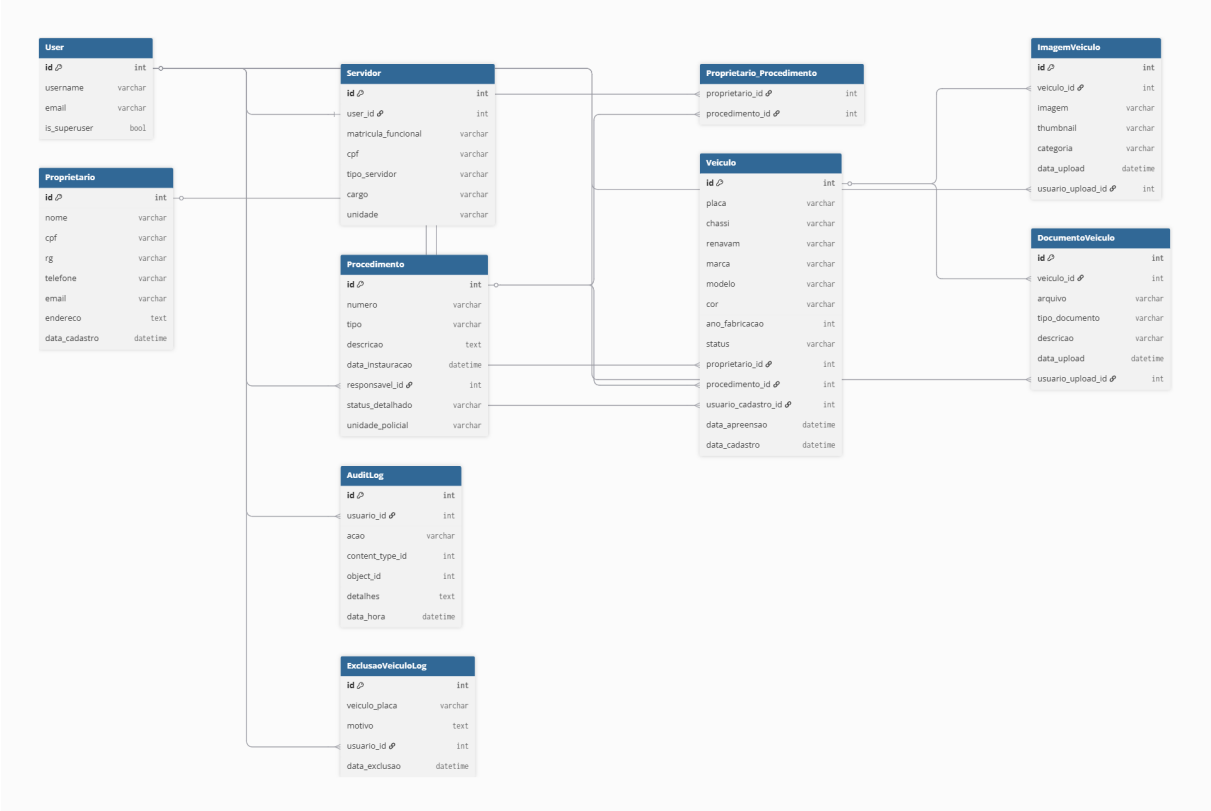

Figura 10 – Diagrama Entidade-Relacionamento do banco de dados do sistema

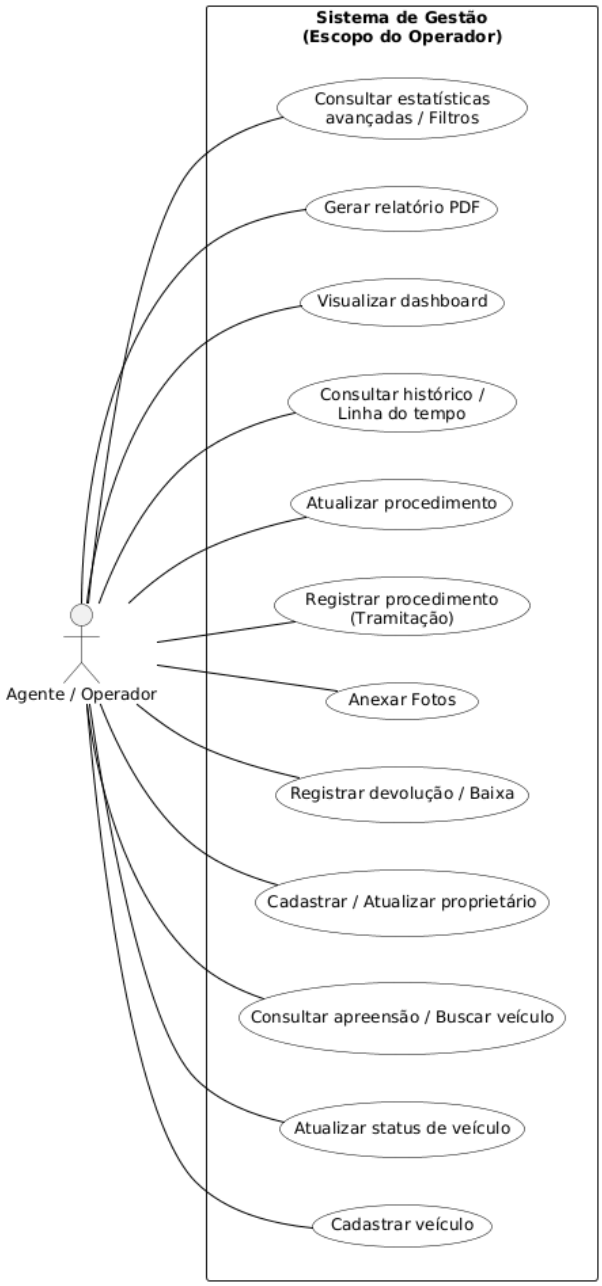

Figura 11 – Diagrama de casos de uso do Operador no sistema de gestão de veículos apreendidos

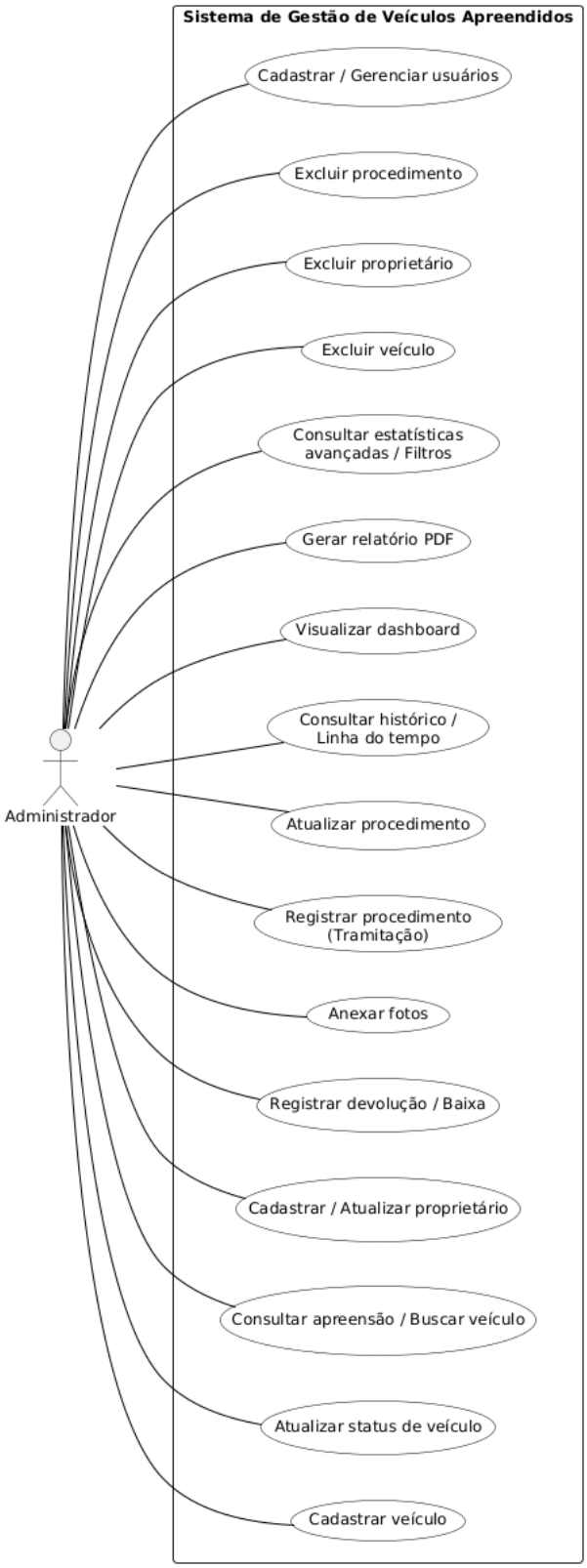

Figura 12 – Diagrama de casos de uso do Administrador no sistema de gestão de veículos apreendidos

Os *Templates* constituem a camada de apresentação do sistema, sendo compostos por arquivos HTML que utilizam as *tags* e filtros da linguagem de templates do Django (`django.template`) para exibição dinâmica dos dados. O sistema faz uso extensivo do módulo `django.template.loader` para carregamento e renderização dos arquivos de template, bem como da diretiva `{% extends %}` para herança de templates, que permite que um template base (`base.html`) defina a estrutura comum de todas as páginas, sendo estendido por templates específicos de cada funcionalidade. Os filtros embutidos, como `{{ campo|date }}` e `{{ campo|default }}`, foram utilizados para formatação e exibição condicional de dados diretamente nas telas. Essa abordagem permite manter a separação entre lógica de negócio e interface, facilitando a manutenção e a evolução visual do sistema. A camada de templates foi integrada ao uso de componentes visuais responsivos, garantindo padronização e melhor experiência do usuário.

3.4.1 Exemplo de Template

Para a implementação da camada de apresentação, foram utilizados templates HTML integrados ao framework Django, os quais permitem a separação entre lógica de negócio e interface gráfica.

```
1  {% extends 'base.html' %}
2
3  {% block content %}
4  <table class="table table-striped">
5      <thead>
6          <tr>
7              <th>Placa</th>
8              <th>Veiculo</th>
9              <th>Status</th>
10         </tr>
11     </thead>
12     <tbody>
13         {% for veiculo in page_obj.object_list %}
14         <tr>
15             <td>{{ veiculo.placa }}</td>
16             <td>{{ veiculo.marca }} {{ veiculo.modelo }}</td>
17             <td>{{ veiculo.get_status_display }}</td>
18         </tr>
19         {% endfor %}
20     </tbody>
21 </table>
22 {% endblock %}
23
```

Figura 13 – Trecho de template Django utilizado na listagem de veículos apreendidos

A Figura 13 apresenta um trecho do template Django responsável pela listagem de veículos apreendidos. O código ilustra dois recursos centrais do sistema de templates do Django: a herança de templates por meio da diretiva `{% extends 'base.html' %}`,

que reutiliza o layout base da aplicação, e a definição de blocos dinâmicos com `{% block content %}`. A iteração `{% for veiculo in page_obj.object_list %}` percorre os registros paginados fornecidos pela View, renderizando dinamicamente os campos placa, marca, modelo e status de cada veículo em linhas de uma tabela HTML. Esse trecho exemplifica na prática a separação entre lógica de negócio e apresentação prevista no padrão MVT: a View processa e filtra os dados, enquanto o template limita-se à sua exibição, sem conter lógica de controle.

A Figura 14 exibe a tela inicial do sistema, apresentada logo após o login bem-sucedido, com acesso direto às principais funcionalidades. A Figura 15 apresenta a tela de listagem de veículos, com mecanismos de busca e filtro por placa, status, unidade e período de apreensão.

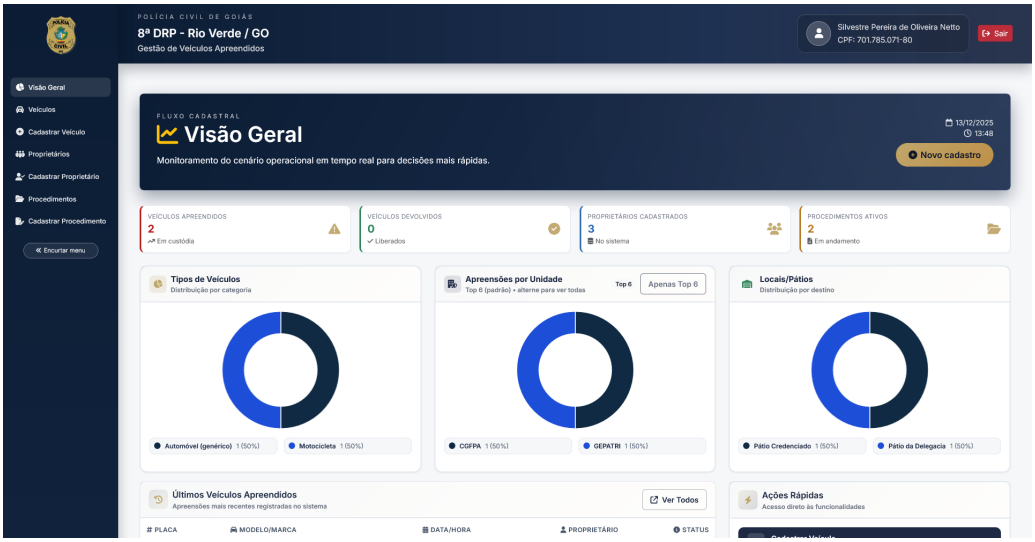

Figura 14 – Tela inicial do Sistema de Gestão de Veículos Apreendidos

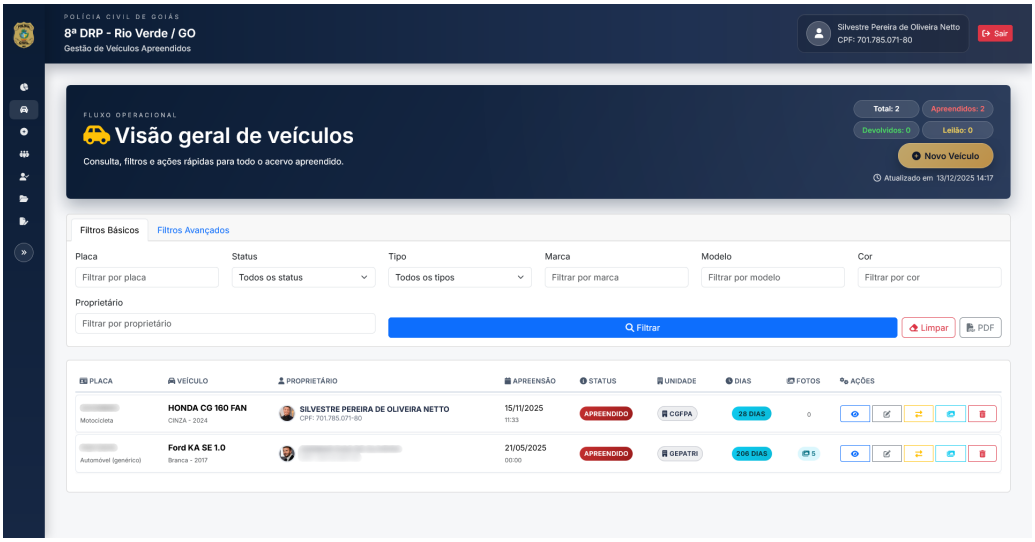

Figura 15 – Tela de listagem de veículos do Sistema de Gestão de Veículos Apreendidos

A aplicação segue a estrutura típica de um projeto Django, organizada em um aplicativo principal responsável pela lógica de gestão, integrado a um projeto maior que contém arquivos de configuração, roteamento, gerenciamento de dependências e inicialização da aplicação. A raiz do projeto contém o arquivo de gerenciamento da aplicação,

enquanto pastas específicas concentram templates, arquivos estáticos (CSS e JavaScript), arquivos de mídia enviados pelos usuários e scripts auxiliares, conforme ilustra a Figura 16.

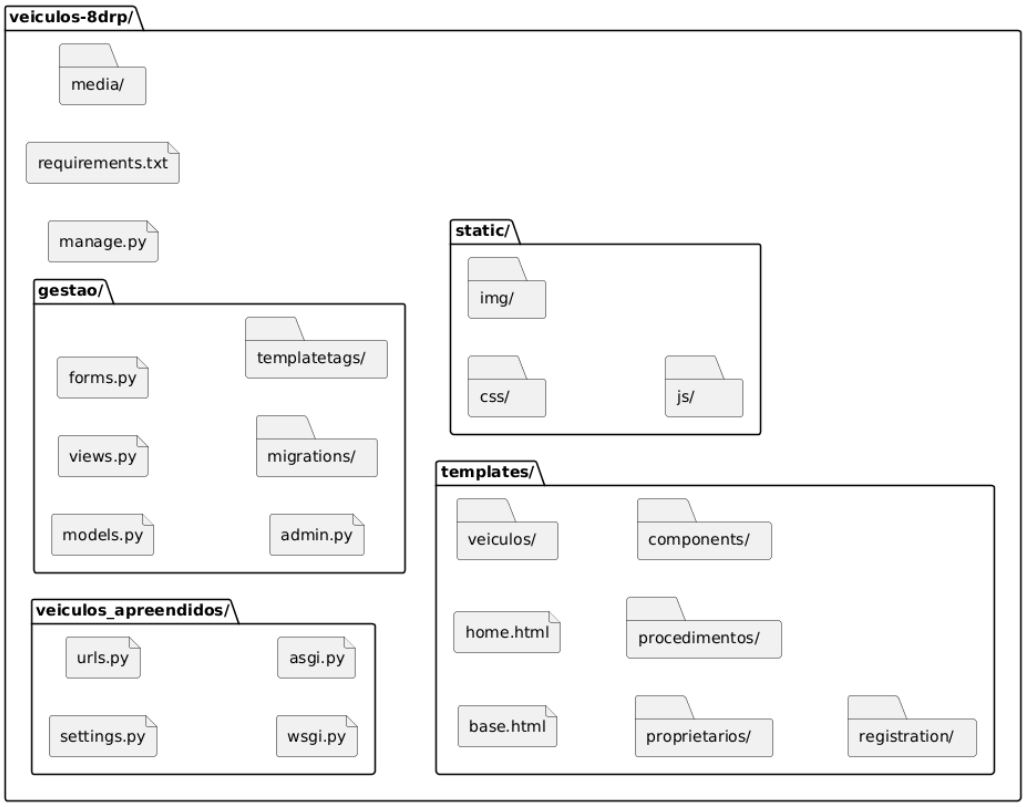

Figura 16 – Estrutura do Projeto

Para otimizar o desempenho do sistema, especialmente em telas de listagem e painéis de indicadores, foram utilizados mecanismos de paginação nativos do Django e estratégias de cache para reduzir consultas repetitivas ao banco de dados. O sistema também emprega recursos de autenticação e controle de acesso, restringindo funcionalidades conforme o perfil do usuário, além de mensagens de feedback para informar o resultado das operações realizadas.

A geração de documentos em formato PDF foi implementada diretamente no backend por meio da biblioteca ReportLab. Essa abordagem possibilitou a criação programática de relatórios administrativos, com controle detalhado sobre layout, tabelas, estilos e formatação, atendendo às necessidades de impressão e arquivamento oficial. O processo consiste na construção do documento em memória, com posterior envio ao navegador do usuário para visualização ou download, conforme exemplificado na Figura 17.

Os arquivos estáticos da aplicação são organizados em diretórios próprios e preparados para ambientes produtivos, enquanto os arquivos de mídia, como imagens associadas aos veículos e documentos anexos, são armazenados separadamente, garantindo melhor organização e controle.

A arquitetura adotada permite que o sistema evolua de forma gradual, mantendo clareza na separação de responsabilidades entre camadas, reutilização de modelos de dados e facilidade de manutenção. A utilização do padrão MVT aliada a boas práticas de organização de código e persistência de dados contribui para a robustez da solução e sua adequação ao contexto institucional da Polícia Civil.

Nos pontos indicados ao longo do trabalho, são apresentados diagramas de casos de uso e diagramas de classes, que complementam a descrição textual da arquitetura,

Figura 17 – Exemplo de relatório administrativo em PDF gerado pelo sistema

ilustrando as interações entre usuários e sistema, bem como os relacionamentos entre as principais entidades modeladas.

3.5 Fluxo Operacional do Sistema

O fluxo operacional do sistema de Gestão de Veículos Apreendidos foi estruturado com base nos procedimentos reais adotados pela unidade policial, buscando representar fielmente as etapas envolvidas no controle, acompanhamento e documentação das apreensões. A definição desse fluxo teve como objetivo garantir simplicidade de uso, padronização das informações e redução de falhas decorrentes de processos manuais.

O acesso ao sistema ocorre por meio de autenticação, assegurando que apenas usuários previamente autorizados possam utilizar suas funcionalidades. Após o login, o usuário é direcionado ao painel principal, onde são apresentadas informações consolidadas sobre o estado atual do sistema, como a quantidade de veículos apreendidos, devolvidos, procedimentos em andamento e outros indicadores relevantes para a gestão administrativa. Esse painel atua como ponto central de navegação, facilitando o acesso rápido às principais funcionalidades.

A partir do painel, o usuário pode iniciar o cadastro de um veículo apreendido, registrando informações essenciais como identificação do veículo, dados do proprietário, local e data da apreensão, condição do bem, checklist de documentos e chaves, bem como o local de destino ou pátio responsável. Durante essa etapa, o sistema realiza validações dos dados informados, assegurando consistência e integridade antes do armazenamento no banco de dados.

Após o cadastro, os veículos passam a integrar a base de dados do sistema e podem ser consultados por meio de mecanismos de busca e filtragem. As funcionalidades de consulta permitem localizar registros a partir de critérios como placa, período de apreensão, status, unidade responsável ou condição do veículo. Para melhorar o desempenho e a usabilidade, as listagens utilizam paginação, evitando a sobrecarga de informações na interface e facilitando a navegação.

O fluxo operacional também contempla a atualização do status dos veículos ao longo do tempo, permitindo o registro de mudanças como transferência de local, andamento de procedimentos administrativos ou devolução ao proprietário. Essas atualizações

garantem o acompanhamento histórico das apreensões, contribuindo para maior controle e rastreabilidade das informações.

Outro elemento central do fluxo é a geração de relatórios administrativos. O sistema possibilita a seleção de filtros específicos para consolidar dados conforme a necessidade do usuário, como listas de veículos por período, tipo ou situação. A partir dessas informações, são gerados documentos em formato PDF, produzidos diretamente no backend, adequados para impressão e arquivamento oficial. Essa funcionalidade reduz o retrabalho e elimina a necessidade de elaboração manual de relatórios.

Durante todo o fluxo operacional, o sistema fornece mensagens de feedback ao usuário, indicando o sucesso ou falha das operações realizadas, como cadastros, atualizações ou geração de documentos. Esse mecanismo contribui para uma interação mais clara e segura, minimizando erros operacionais.

O fluxo foi projetado de forma integrada às camadas do sistema, garantindo que todas as operações realizadas na interface reflitam diretamente nos dados persistidos no banco de dados. Essa integração assegura que as informações apresentadas em consultas, dashboards e relatórios sejam consistentes e atualizadas, fortalecendo a confiabilidade do sistema no contexto institucional.

A organização do fluxo operacional adotado permite que o sistema seja utilizado de maneira intuitiva, mesmo por usuários com pouca familiaridade com sistemas informatizados, ao mesmo tempo em que oferece suporte adequado às demandas administrativas da Polícia Civil. Dessa forma, o fluxo operacional contribui diretamente para a eficiência do gerenciamento de veículos apreendidos, alinhando tecnologia e prática institucional. A Figura 18 ilustra o diagrama de atividades correspondente ao fluxo operacional descrito.

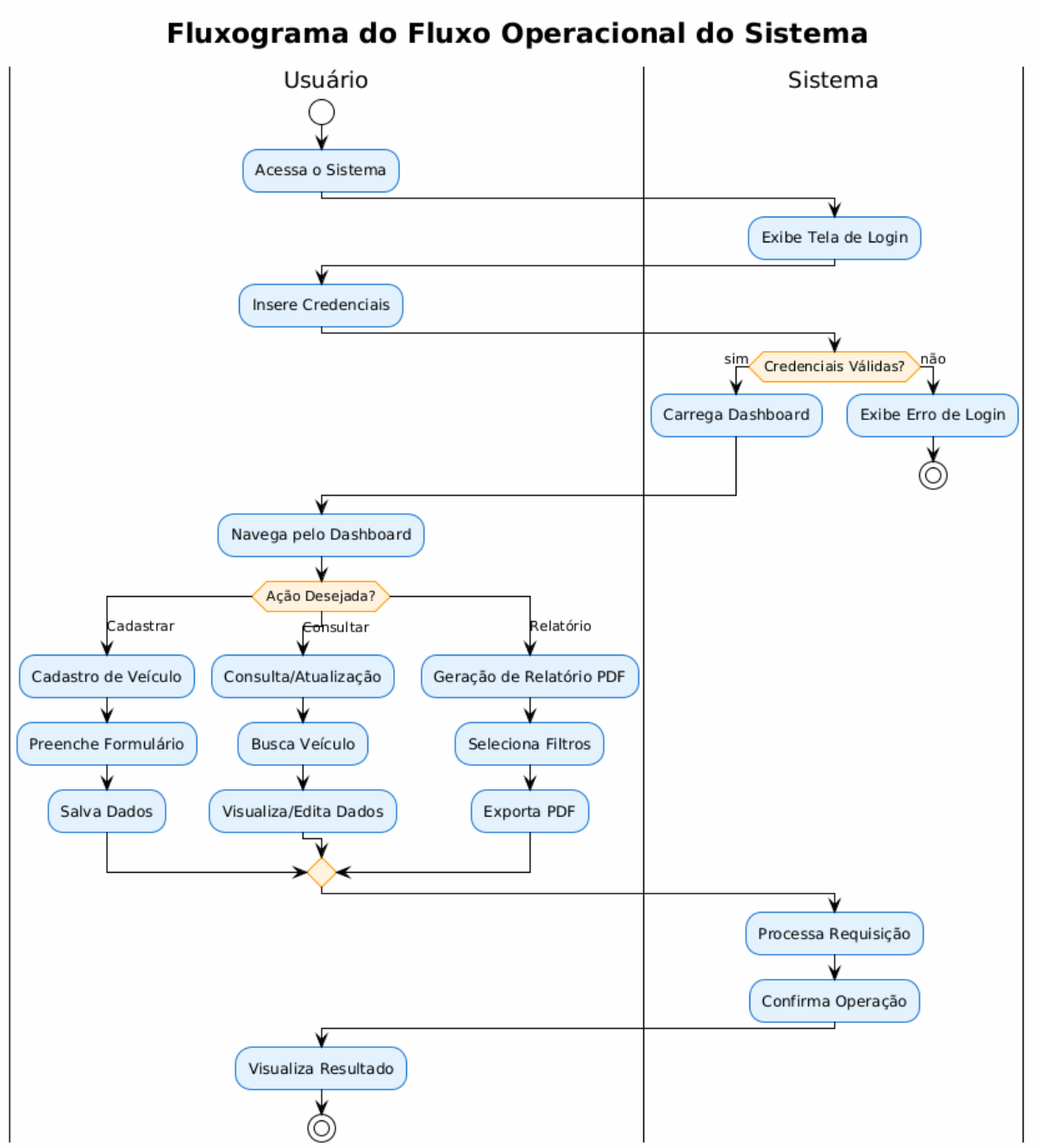

Figura 18 – Diagrama de Atividades do Fluxo Operacional do Sistema

3.6 Organização em Camadas e Manutenibilidade

A arquitetura apresentada fundamenta-se na separação clara de responsabilidades entre as camadas do sistema, o que contribui diretamente para a organização do código e para sua manutenibilidade. A divisão entre dados, lógica de aplicação e apresentação permite que cada parte do sistema evolua de forma independente, reduzindo o acoplamento entre componentes.

Essa organização facilita a realização de correções e ajustes pontuais, uma vez que alterações em regras de negócio ou na interface do usuário podem ser implementadas sem impacto direto nas demais camadas. Do mesmo modo, a reutilização dos modelos de dados em diferentes funcionalidades do sistema assegura consistência das informações e reduz a duplicidade de código.

A estrutura em camadas também favorece a evolução futura da aplicação, permitindo a ampliação de funcionalidades ou adaptações a novos requisitos institucionais sem a necessidade de reestruturação completa do sistema. Dessa forma, a arquitetura adotada contribui para a longevidade da solução e sua adequação às demandas operacionais da Polícia Civil, conforme representado na Figura 19.

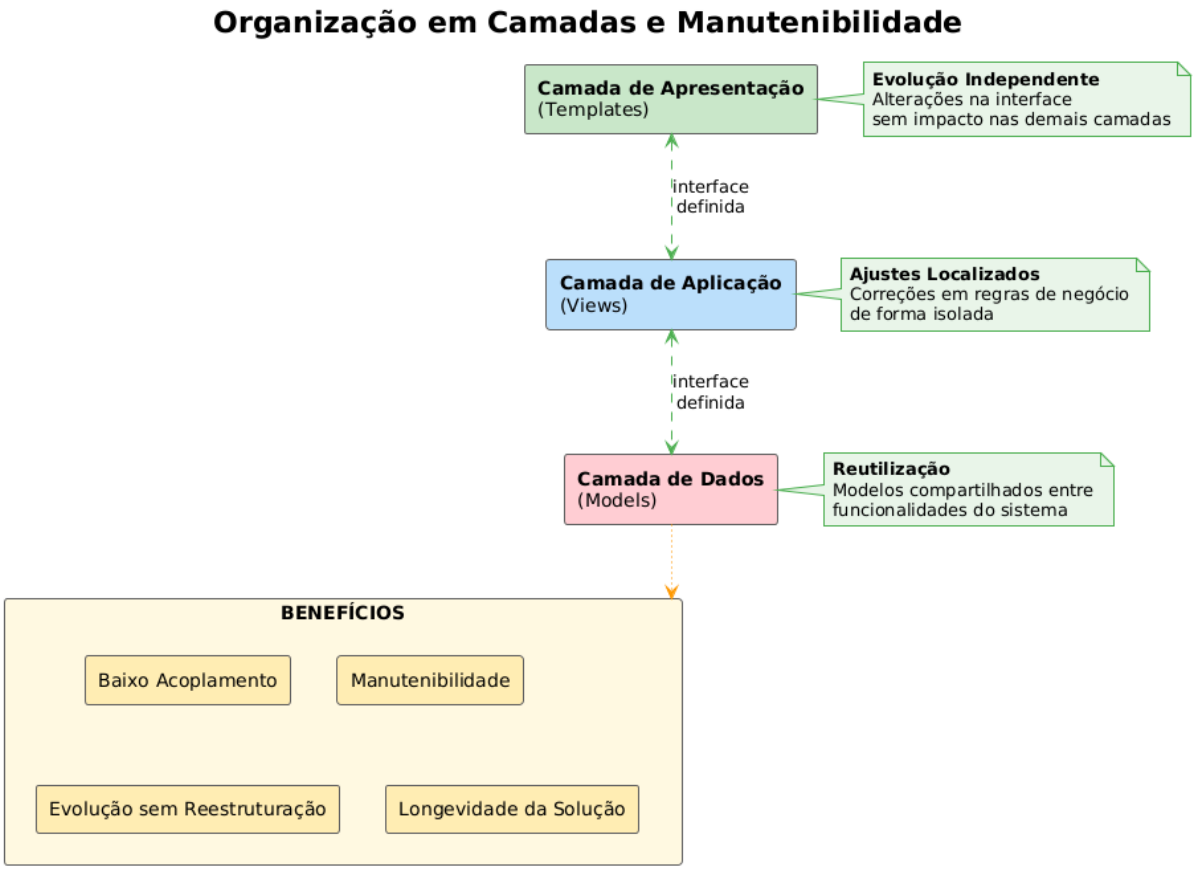

Figura 19 – Organização em Camadas e Manutenibilidade

3.6.1 Segurança e Controle de Acesso

O sistema de gestão de veículos apreendidos lida com informações sensíveis e, por isso, incorpora diversos mecanismos de segurança e controle de acesso para garantir a confidencialidade, a integridade e a rastreabilidade dos dados.

Autenticação e gestão de sessão: A autenticação de usuários baseia-se no mecanismo nativo do Django: as credenciais são validadas com o formulário de autenticação padrão (`AuthenticationForm` do módulo `django.contrib.auth.forms`), e o usuário é autenticado via função `authenticate()` antes de iniciar a sessão. As senhas são armazenadas de forma criptografada usando o algoritmo PBKDF2 com SHA-256 por padrão, conforme a infraestrutura padrão do *framework*. A configuração do projeto define validadores de senha (comprimento mínimo de oito caracteres, detecção de senhas similares ao nome de usuário e rejeição de senhas puramente numéricas), a expiração automática da sessão após 30 minutos de inatividade (`SESSION_COOKIE_AGE = 1800`) e o encerramento automático ao fechar o navegador (`SESSION_EXPIRE_AT_BROWSER_CLOSE = True`). A finalização da sessão é tratada em *view* específica que utiliza `login_required` para restringir o acesso e efetua o *logout* com confirmação visual, conforme ilustra a Figura 20.

Proteção contra vulnerabilidades web: O conjunto de *middlewares* do Django inclui proteção automática contra CSRF (`CsrfViewMiddleware`), cabeçalhos de segurança HTTP (`SecurityMiddleware`) e clique-jacking (`XFrameOptionsMiddleware`). O uso exclusivo do ORM para interações com o banco de dados previne injeção de SQL; os templates do Django realizam escape automático de dados exibidos, prevenindo *Cross-Site Scripting* (XSS).

Controle de acesso por perfil: O modelo *Servidor*, que estende o usuário padrão do Django com dados funcionais, permite classificar o tipo de servidor (policial ou administrativo). As funções críticas de exclusão de veículos, proprietários e procedimentos só podem ser executadas por usuários com privilégio de superusuário (`is_superuser`). Nesses casos, a *view* solicita a senha do administrador, valida novamente o usuário via `authenticate()` e verifica a propriedade `is_superuser` antes de prosseguir. A exclusão é registrada em *log* mediante o modelo *ExclusaoVeiculoLog*, que armazena a identificação do veículo, o motivo da exclusão, o usuário responsável e a data e hora do evento, conforme ilustrado na Figura 21. Esse registro favorece a auditoria e a responsabilização institucional.

Estabilidade e controle de arquivos: As *views* responsáveis pelo *upload* de imagens e documentos são protegidas por `login_required`, garantindo que apenas usuários autenticados possam anexar ou remover arquivos. Os formulários impõem validação de extensão e tamanho máximo (5 MB para imagens e 20 MB para documentos), o que restringe a possibilidade de inserção de arquivos maliciosos e previne sobrecarga do sistema de armazenamento. O campo *usuario_upload* registra quem realizou o envio, contribuindo para a rastreabilidade. A execução da aplicação em contêineres Docker garante isolamento do ambiente e reprodutibilidade da implantação, reduzindo riscos de instabilidade por incompatibilidades de dependências entre ambientes de desenvolvimento e produção.

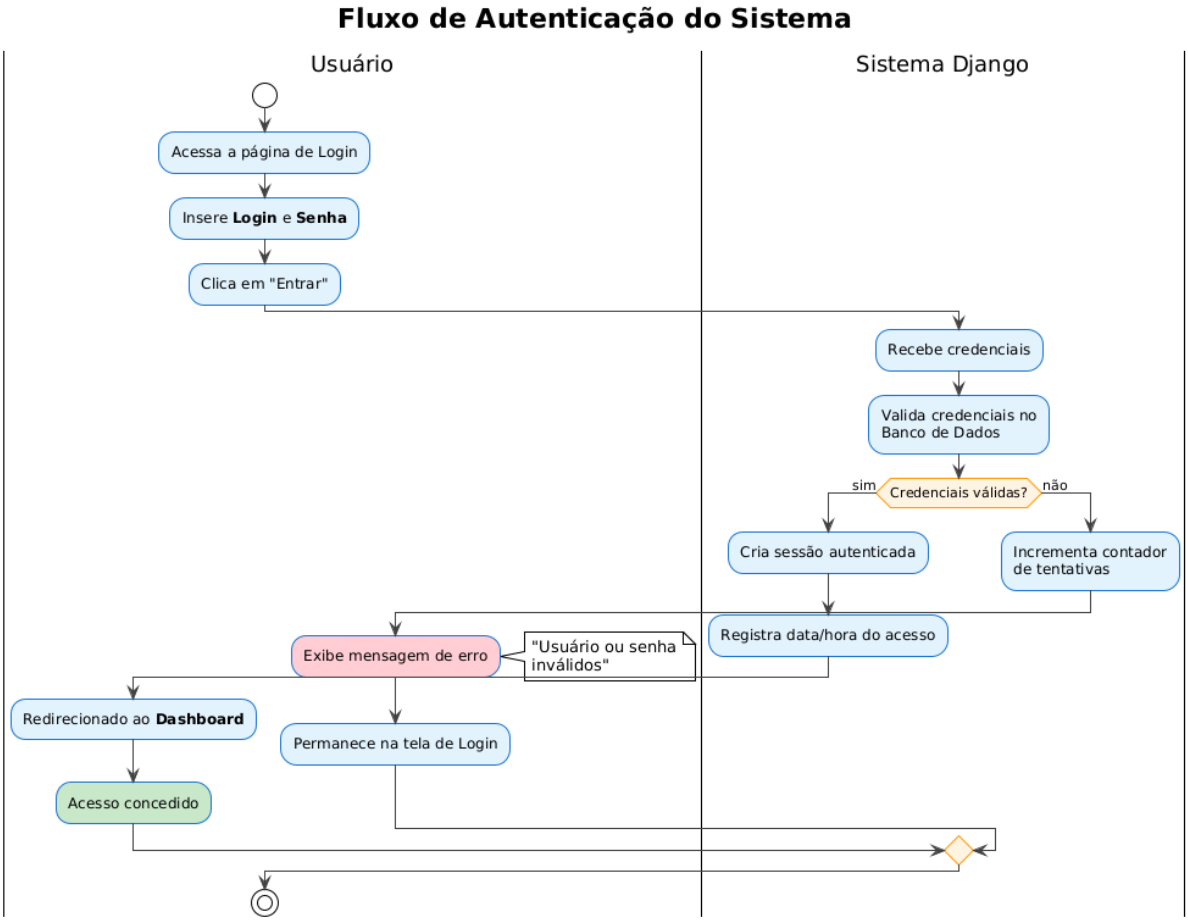

Figura 20 – Diagrama do Fluxo de Autenticação do Sistema

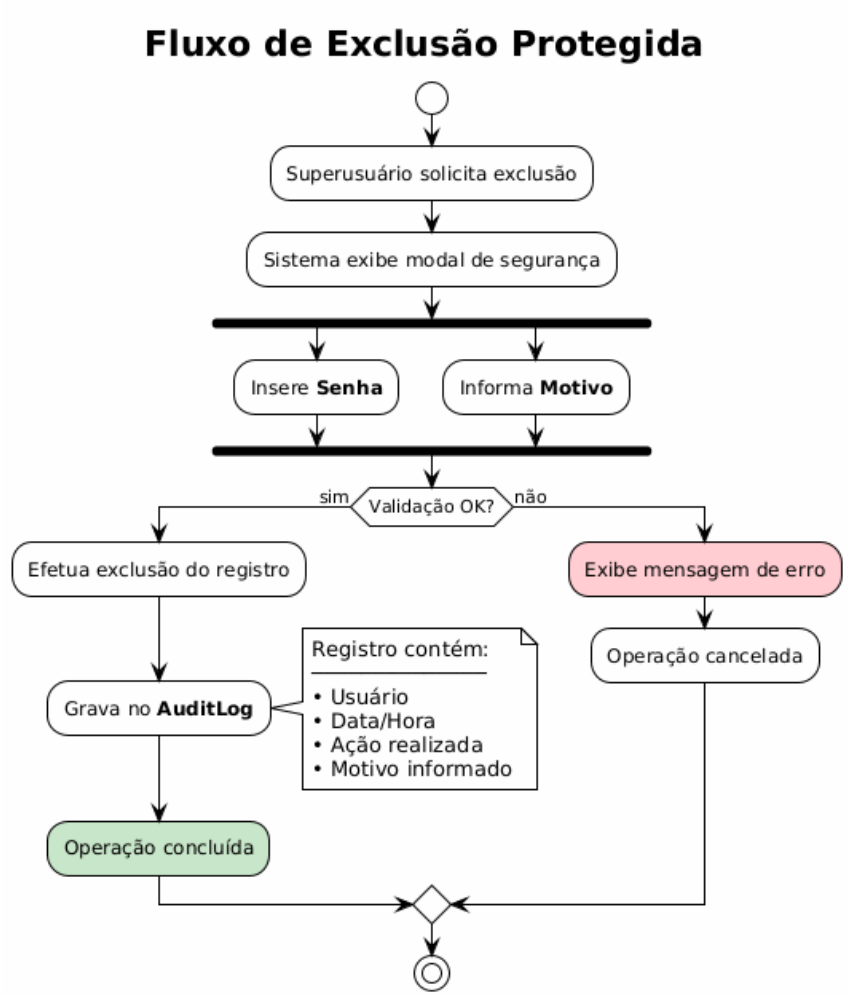

Figura 21 – Mecanismo de Exclusão Protegida com Registro em Log de Auditoria

4 RESULTADO E DISCUSSÕES

Após a conclusão do desenvolvimento, o sistema de gestão de veículos apreendidos foi disponibilizado para avaliação e uso pelos servidores da 8ª Delegacia Regional de Polícia Civil de Goiás (8ª DRP), unidade responsável pela custódia de bens apreendidos. Essa etapa teve como objetivo analisar, na prática, os ganhos obtidos com a informatização do processo, assim como identificar oportunidades de aprimoramento.

4.1 Avaliação com usuários

4.1.1 Metodologia de avaliação

Para validar a aplicabilidade do sistema, os testes foram conduzidos diretamente nas dependências da 8ª DRP, em ambiente real de uso, com a participação de **6 servidores**: 3 policiais civis responsáveis pelo registro operacional de apreensões e 3 servidores administrativos envolvidos na geração de relatórios e no controle de acesso. Os participantes não possuíam experiência prévia com o sistema, o que permitiu avaliar a curva de aprendizado em condições reais.

As sessões foram realizadas em **2 encontros presenciais** com duração média de **2 horas cada**, totalizando aproximadamente 4 horas de avaliação. No primeiro encontro, os participantes receberam uma breve apresentação das funcionalidades principais, sem treinamento formal. Em seguida, foram orientados a executar livremente as tarefas propostas nos cenários de teste. No segundo encontro, foram avaliados o uso autônomo do sistema e o refinamento de funcionalidades a partir das sugestões coletadas na sessão anterior.

Para avaliar a usabilidade do sistema de forma objetiva, foram definidas as seguintes **métricas de avaliação**:

- **Taxa de conclusão de tarefas:** proporção de cenários executados com sucesso pelos participantes, sem a necessidade de intervenção externa ou instrução adicional após a apresentação inicial. Meta estabelecida: $\geq 90\%$;
- **Tempo de execução das tarefas:** medição do tempo necessário para que cada participante concluísse as tarefas principais (cadastro de veículo, busca por placa e geração de relatório PDF). Meta estabelecida: inferior a 5 minutos por tarefa;
- **Tempo de resposta do sistema:** tempo decorrido entre a ação do usuário e a exibição do resultado pelo sistema, incluindo consultas e listagens. Meta estabelecida: inferior a 3 segundos;
- **Ausência de erros críticos:** nenhum cenário de teste deve resultar em perda, inconsistência ou corrompimento de dados.

Os **critérios de aceitação** adotados para validação de cada cenário foram:

- Execução bem-sucedida da tarefa dentro das condições esperadas, sem erros críticos ou bloqueios;
- Interface compreensível sem necessidade de instrução adicional após a apresentação inicial;
- Tempo de resposta adequado ao fluxo operacional (inferior a 3 segundos para consultas e listagens);
- Ausência de perda ou inconsistência de dados durante as operações realizadas.

Os resultados de cada cenário foram registrados como **Aprovado** ou **Reprovado**, conforme a capacidade do sistema de atender ao cenário dentro dos critérios definidos.

Todos os 10 cenários executados foram aprovados na avaliação final, atingindo taxa de conclusão de 100% — acima da meta estabelecida —, conforme detalhado nas Tabelas 2 e 3 da seção de Materiais e Métodos. Nenhum cenário registrou tempo de resposta superior a 3 segundos nas operações de consulta e listagem.

4.1.2 Resultados da avaliação

Os resultados apresentados a seguir refletem a percepção dos usuários participantes das sessões de avaliação, coletada por meio de observação direta, registro de desempenho nas tarefas e comentários informais durante e após os encontros. Por se tratar de uma avaliação qualitativa com usuários reais em ambiente institucional, os dados quantitativos (como taxas de conclusão e tempos de resposta) foram observados durante os testes, enquanto as percepções sobre usabilidade e facilidade de uso foram relatadas pelos próprios servidores.

Para validar a aplicabilidade do sistema, foram realizados testes em ambiente real com a participação de agentes e administrativos da 8ª DRP. Os usuários receberam treinamento básico sobre o fluxo de cadastro e consulta, e posteriormente realizaram operações cotidianas (registro de apreensões, atualização de status, geração de documentos e consultas de veículos e procedimentos). Durante a avaliação, observou-se que o tempo de aprendizagem foi reduzido, uma vez que a interface faz uso de componentes conhecidos (menus, formulários, filtros e tabelas), e que os usuários conseguiram navegar de forma intuitiva pelas funcionalidades. Na percepção dos participantes, relatada informalmente durante as sessões, a facilidade para localizar dados específicos e a clareza na exibição das informações foram os aspectos mais valorizados, atribuídos à organização das telas e aos filtros disponíveis. É importante destacar que essas avaliações positivas expressam a percepção subjetiva dos servidores, e não medições objetivas e controladas; a consolidação de dados comparativos mais precisos dependerá do uso contínuo do sistema ao longo do tempo.

4.2 Benefícios observados

Os benefícios descritos nesta seção foram identificados a partir dos relatos e da percepção dos servidores participantes das sessões de avaliação, e não de medições quantitativas comparativas formais. A base de evidências é qualitativa e fundamentada na experiência direta dos usuários com o sistema em substituição aos métodos manuais anteriores.

Um dos principais ganhos relatados pelos servidores foi a **centralização das informações**. Antes da implantação, o controle de veículos e procedimentos era feito por meio de planilhas e anotações manuais, o que dificultava o acesso e a atualização dos dados. Com a aplicação, os registros passaram a ser armazenados de forma estruturada em banco relacional, eliminando inconsistências e duplicidades. A funcionalidade de *filtros avançados* permitiu consultas rápidas por placa, status, período de apreensão, unidade responsável, tipo de veículo, entre outros critérios, reduzindo significativamente o tempo necessário para encontrar um registro.

Outra melhoria importante refere-se à **vinculação entre veículos e procedimentos**. Por meio dos relacionamentos definidos no modelo de dados, cada veículo registrado pode ser associado a um ou mais procedimentos, e essa associação é exibida de forma clara na interface. Os servidores relataram que essa funcionalidade facilitou a busca de procedimentos relacionados a determinado veículo (e vice-versa), permitindo acompanhamento

mais preciso dos trâmites e prazos. Da mesma forma, as telas de listagem com paginação e ordenação tornaram a navegação por grandes conjuntos de registros de veículos e procedimentos mais eficiente.

Com a adoção do sistema, a **emissão de relatórios** também se tornou mais ágil. A geração automática de PDFs com listas de veículos, estatísticas por tipo ou unidade e histórico de apreensões eliminou a necessidade de compor relatórios manualmente. Os relatórios são gerados de acordo com os filtros aplicados pelo usuário, garantindo que apenas as informações relevantes sejam incluídas.

4.3 Adoção e uso contínuo

Desde a disponibilização do sistema, a equipe da 8ª DRP incorporou progressivamente a plataforma à rotina operacional. Novas apreensões passaram a ser cadastradas diretamente na plataforma, e os registros são atualizados à medida que os veículos são devolvidos, encaminhados para leilão ou descartados. O processo de integração da ferramenta ao fluxo de trabalho transcorreu sem resistência significativa, em parte pela familiaridade dos servidores com interfaces baseadas em navegador web e pela consistência dos componentes visuais adotados.

O treinamento dos usuários foi conduzido de forma informal entre os próprios colegas já habituados ao sistema, o que evidenciou que a curva de aprendizado é baixa: a maioria dos servidores tornou-se operacionalmente independente em poucas sessões. Esse modelo de disseminação espontânea indica que a adoção não depende de capacitações formais extensas, representando um fator favorável à expansão do uso para outras unidades.

Por encontrar-se em fase inicial de implantação, dados quantitativos comparativos consolidados ainda estão sendo coletados ao longo do uso contínuo. O engajamento crescente da equipe e a substituição efetiva dos controles manuais pelas operações na plataforma demonstram, no entanto, que o sistema foi assimilado como ferramenta principal de gestão da unidade, e não apenas como recurso auxiliar.

4.4 Discussão e perspectivas futuras

Os resultados obtidos demonstram que o sistema atendeu às principais expectativas de informatização e controle, permitindo maior transparência e eficiência no gerenciamento de veículos apreendidos. Entretanto, durante a avaliação foram sugeridas algumas melhorias. Entre elas, destacam-se a possibilidade de **integração com bancos de dados externos** (por exemplo, consultas automáticas a informações do DETRAN ou RENAVAM) para validar dados cadastrais, e a criação de um **módulo de notificação** que alerte os usuários sobre prazos de procedimentos ou vencimento de documentos.

Em síntese, o uso do sistema na 8ª DRP evidenciou uma evolução substancial em relação aos métodos manuais empregados anteriormente. A facilidade de uso, a rapidez nas consultas e a redução de erros de registro apontam para um ganho efetivo de produtividade. A continuidade do monitoramento e o atendimento das sugestões de melhoria contribuirão para consolidar o sistema como ferramenta indispensável para o controle de veículos apreendidos na esfera policial.

4.5 Registro de Programa de Computador junto ao INPI

Como resultado do desenvolvimento deste trabalho, dois módulos do sistema foram registrados formalmente junto ao Instituto Nacional da Propriedade Industrial

(INPI), em conformidade com a Lei nº 9.609, de 19 de fevereiro de 1998, que dispõe sobre a proteção da propriedade intelectual de programas de computador no Brasil. Os depósitos foram realizados em novembro de 2025 e os certificados expedidos em dezembro do mesmo ano, tendo o Instituto Federal de Educação, Ciência e Tecnologia Goiano como titular. Os autores registrados são: Adriano Soares de Oliveira Bailão, Gustavo Moura Barros, Vitor Diniz Dantas, Silvestre Pereira de Oliveira Netto e Andréa Barboza Proto Sardi. Os dados formais de ambos os certificados — incluindo número de registro, título do programa e data de depósito — estão consolidados na Tabela 6.

Tabela 6 – Registros de Programa de Computador obtidos junto ao INPI

Campo	**Backend**	**Frontend**
Processo Nº	BR512025006985-5	BR512025006723-2
Título	Plataforma de Backend em Django (MTV) para Gestão de Veículos Apreendidos, Baixas e Evidências	Frontend Web Responsivo com Django Templates, Bootstrap e Chart.js para Gestão de Veículos Apreendidos, Baixas e Evidências
Linguagem	Python	HTML, JavaScript, CSS
Publicação	08/12/2025	10/12/2025
Expedição	30/12/2025	23/12/2025
Titular	Instituto Federal de Educação, Ciência e Tecnologia Goiano	

Fonte: INPI (2025).

A Figura 22 e a Figura 23 reproduzem os certificados originais expedidos pelo INPI. O registro garante proteção legal por 50 anos a partir das respectivas datas de publicação, conforme estabelecido na legislação vigente. A obtenção desses certificados representa um resultado que supera o escopo típico de trabalhos de conclusão de curso, conferindo ao software desenvolvido o reconhecimento formal do Estado como produto de propriedade intelectual.

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA, COMÉRCIO E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512025006985-5**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1° de janeiro subsequente à data de 08/12/2025, em conformidade com o §2°, art. 2° da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Plataforma de Backend em Django (MTV) para Gestão de Veículos Apreendidos, Baixas e Evidências

Data de publicação: 08/12/2025

Data de criação: 11/11/2025

Titular(es): INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Autor(es): ADRIANO SOARES DE OLIVEIRA BAILÃO; GUSTAVO MOURA BARROS; VITOR DINIZ DANTAS; SILVESTRE PEREIRA DE OLIVEIRA NETTO; ANDRÉA BARBOZA PROTO SARDI

Linguagem: PYTHON

Campo de aplicação: AD-04

Tipo de programa: AP-01

Algoritmo hash: SHA-256

Resumo digital hash: e67ee875e32992f4d8c0d34bdde089cf4b13d94858914d4aa2b955e496546518

Expedido em: 30/12/2025

Aprovado por:
ERICA GUIMARAES CORREA
Chefe da DIPTO

Figura 22 – Certificado de Registro – Backend (processo BR512025006985-5)

REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA, COMÉRCIO E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512025006723-2**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1° de janeiro subsequente à data de 10/12/2025, em conformidade com o §2°, art. 2° da Lei 9.609, de 19 de Fevereiro de 1998.

Título: Frontend Web Responsivo com Django Templates, Bootstrap e Chart.js para Gestão de Veículos Apreendidos, Baixas e Evidências

Data de publicação: 10/12/2025

Data de criação: 11/11/2025

Titular(es): INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA GOIANO

Autor(es): ADRIANO SOARES DE OLIVEIRA BAILÃO; GUSTAVO MOURA BARROS; VITOR DINIZ DANTAS; SILVESTRE PEREIRA DE OLIVEIRA NETTO; ANDRÉA BARBOZA PROTO SARDI

Linguagem: HTML; JAVA SCRIPT; CSS

Campo de aplicação: AD-04

Tipo de programa: AP-01

Algoritmo hash: SHA-256

Resumo digital hash: 385f0f6ce4dc725183e8f53efa923d7d25bb9d718ccfd83707adb62793eed886

Expedido em: 23/12/2025

Aprovado por:
ERICA GUIMARAES CORREA
Chefe da DIPTO

Figura 23 – Certificado de Registro – Frontend (processo BR512025006723-2)

Fonte: INPI (2025).

5 CONCLUSÃO

O sistema de gestão de veículos apreendidos desenvolvido ao longo deste trabalho tinha como principal objetivo substituir o controle manual baseado em planilhas e documentos físicos por uma solução informatizada capaz de centralizar dados, agilizar consultas e aumentar a transparência dos processos. A escolha de tecnologias de código aberto, como Python, o framework Django e o banco de dados PostgreSQL, permitiu a construção de uma aplicação robusta e de fácil manutenção, alinhada às necessidades institucionais da 8ª Delegacia Regional de Polícia Civil do Estado de Goiás.

A avaliação realizada com os servidores da Delegacia demonstrou que a ferramenta cumpriu seus propósitos. Os usuários ressaltaram a facilidade de uso e a rapidez na obtenção de informações, destacando a eliminação de redundâncias e inconsistências que ocorriam no método manual. A funcionalidade de filtros avançados facilitou a busca por veículos e procedimentos, enquanto a vinculação entre registros permitiu acompanhamento preciso de cada caso. Além disso, a geração automática de relatórios em formato PDF simplificou a elaboração de documentos formais, reduzindo o tempo gasto nessas tarefas e aumentando a confiabilidade dos dados.

A implantação do sistema resultou em ganhos tangíveis de produtividade e eficiência para a 8ª DRP, comprovando que a adoção de tecnologias abertas e bem estruturadas, aliada ao envolvimento contínuo dos usuários finais, é capaz de modernizar processos institucionais com impacto direto na qualidade do serviço público prestado. Assim, conclui-se que o projeto atingiu seus objetivos e representa uma contribuição concreta para a digitalização da gestão de veículos apreendidos no âmbito da segurança pública do Estado de Goiás. A relevância institucional do trabalho foi formalmente reconhecida por meio do registro de dois programas de computador junto ao INPI — processo nº BR512025006985-5 (backend em Python/Django) e processo nº BR512025006723-2 (frontend em HTML, JavaScript e CSS) —, ambos com titularidade do IF Goiano e expedidos em dezembro de 2025, assegurando proteção legal de propriedade intelectual por 50 anos.

Mesmo com os ganhos obtidos, algumas limitações permaneceram. A pesquisa foi restrita à 8ª DRP e não avaliou a adoção em outras unidades, o que poderia trazer novas necessidades. Além disso, a integração com sistemas externos (DETRAN, RENAVAM) ainda não foi implementada e a ausência de um módulo de notificação pode impactar o acompanhamento de prazos. Essas limitações, contudo, não invalidam os resultados: elas ressaltam a necessidade de evoluir a solução e mostram que a modernização pode ser contínua.

5.1 Trabalhos Futuros

Apesar dos resultados positivos obtidos, há diversas oportunidades para a expansão e o aprimoramento da solução desenvolvida. Entre as principais vertentes de trabalhos futuros, destacam-se:

- **Possível integração com sistemas externos do DETRAN:** Incorporar serviços de consulta ao banco de dados do Departamento Estadual de Trânsito possibilitaria que, ao digitar uma placa, o sistema recuperasse automaticamente informações oficiais sobre o veículo (marca, modelo, ano e situação de licenciamento). Essa integração eliminaria a necessidade de preencher manualmente dados básicos, reduziria erros de digitação e aumentaria a confiabilidade dos registros.

- **Compartilhamento de informações com outras autoridades:** Desenvolver interfaces de interoperabilidade com órgãos municipais, como a Agência Municipal de Trânsito (AMT), e com outras unidades da Polícia Civil permitiria o intercâmbio seguro de dados sobre veículos apreendidos. Esse compartilhamento facilitaria a fiscalização, reduziria duplicidade de registros e possibilitaria o acompanhamento conjunto de trâmites e procedimentos.
- **Melhorias na usabilidade e mobilidade:** A criação de um módulo adaptado para dispositivos móveis, ou mesmo de um aplicativo específico, ampliaria o alcance do sistema, permitindo que agentes em campo realizem consultas e atualizações diretamente durante as operações. Customizações de interface, como painéis personalizáveis conforme o perfil do usuário, também contribuiriam para a experiência de uso.
- **Análises estatísticas e planejamento:** Explorar os dados armazenados para gerar relatórios analíticos sobre tipos de veículos apreendidos, locais de apreensão, frequência de procedimentos e tempos de tramitação pode apoiar o planejamento operacional da instituição, subsidiar decisões gerenciais e contribuir para a formulação de políticas públicas mais eficientes.

A execução dessas sugestões visa a ampliar o impacto do sistema, tanto em termos de funcionalidade quanto de alcance institucional, consolidando-o como uma ferramenta estratégica para a gestão de veículos apreendidos e servindo de base para iniciativas semelhantes em outras delegacias e órgãos de trânsito.

Referências

BECK, K. et al. *Manifesto for Agile Software Development*. 2001. Disponível em: <https://agilemanifesto.org>. Acesso em: 15 jan. 2025. Citado na página 3.

DATE, C. J. *Introdução a Sistemas de Banco de Dados*. 8. ed. Rio de Janeiro: Elsevier, 2004. Citado na página 4.

Django Software Foundation. *Django Documentation*. 2024. Disponível em: <https://docs.djangoproject.com>. Acesso em: 10 jan. 2025. Citado 2 vezes nas páginas 3 e 4.

FORCIER, J.; BISSEX, P.; CHUN, W. J. *Python Web Development with Django*. Boston: Pearson Education, 2008. Citado na página 5.

GAMMA, E. et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston: Addison-Wesley, 1995. Citado na página 4.

LUTZ, M. *Learning Python*. 5. ed. Sebastopol: O'Reilly Media, 2013. Citado na página 3.

NETO, F. G. C. *SYSFROTA: Um Sistema de Gestão Individualizada de Frota de Veículos*. Monografia (Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas)) — Universidade Federal do Rio Grande do Norte, Escola Agrícola de Jundiaí, Macaíba, 2022. Citado na página 5.

OWASP. *OWASP Top Ten 2021*. 2021. Disponível em: <https://owasp.org/www-project-top-ten/>. Acesso em: 20 jan. 2025. Citado na página 5.

PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software: Uma Abordagem Profissional*. 8. ed. Porto Alegre: AMGH, 2016. Citado na página 3.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. 2020. Disponível em: <https://scrumguides.org/scrum-guide.html>. Acesso em: 10 fev. 2025. Citado na página 3.

SILVA, H. W. d. S.; BARBOSA, W. d. O. *Sistema de Gerenciamento de Veículos Apreendidos para a Polícia Civil do Estado do Pará*. Monografia (Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação)) — Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais, Faculdade de Computação, Belém, 2018. Citado na página 5.

SILVA, R. Q. *Sistema de Informação para Controle de Veículos e Viagens de uma Transportadora*. Monografia (Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação)) — Pontifícia Universidade Católica de Goiás, Escola Politécnica, Goiânia, 2022. Citado na página 5.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Citado na página 3.

VERíSSIMO, L. G. S. *Estudo de Caso: Análise e Projeto do Sistema de Controle de Veículos Apreendidos (SISCOVA)*. Monografia (Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação)) — Universidade Estadual de Goiás, Unidade Universitária de Itaberaí, Itaberaí, 2013. Citado na página 5.