

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

AMANDA KAROLINE CAROLINO DA SILVA

**UM SISTEMA DE APOIO À COLETA DE DADOS EM PERÍCIAS CRIMINAIS DE
VEÍCULOS**

GUARAPUAVA

2022

AMANDA KAROLINE CAROLINO DA SILVA

**UM SISTEMA DE APOIO À COLETA DE DADOS EM PERÍCIAS CRIMINAIS DE
VEÍCULOS**

**A SYSTEM TO SUPPORT DATA COLLECTION IN VEHICLE CRIMINAL
EXPERTISE**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Andres Jessé Porfirio

GUARAPUAVA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

AMANDA KAROLINE CAROLINO DA SILVA

UM SISTEMA DE APOIO À COLETA DE DADOS EM PERÍCIAS CRIMINAIS DE VEÍCULOS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Sistemas para Internet do Curso de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná (UTFPR).

Data da aprovação: 20/junho/2022

Prof. Andres Jessé Porfírio
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Diego Marczal
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

Prof. Guilherme da Costa Silva
Mestre
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

GUARAPUAVA
2022

Dedico este trabalho à minha família e amigos
que sempre me apoiaram.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela oportunidade de completar mais esta etapa da minha vida.

À minha família, em especial aos meus pais Jacqueline Dagmar e Gilmar Carolino, que nunca mediram esforços para proporcionar a educação necessária para minha formação profissional. Aos meus irmãos e minhas maiores referências não só na programação, mas na vida, Ruhan Carolino, Rehnan Carolino, Hudson Carolino e Ananda Zahara, pelas palavras de incentivo, pela dádiva de nossa amizade e união, que mesmo longe sempre estiveram comigo. Amo vocês.

Aos meus amigos e colegas, por compreender a minha ausência nos últimos meses. Em especial a Eduarda Lara e o meu grande amigo Renan Gabriel que esteve ao meu lado desde o início dessa jornada, incentivando, ajudando a vencer cada obstáculo.

Aos meus professores da Universidade Tecnológica Federal do Paraná, em especial ao meu orientador Prof. Dr. Andres Jessé Porfirio, pelo incentivo, por acreditar em mim, pela dedicação e paciência durante a realização deste trabalho.

A todos, muito obrigada!

"O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis."(Alencar, José).

RESUMO

A Perícia Criminal, área de muita importância para o processo judicial, tem o encargo de realizar exames com objetivo de esclarecer fatos delituosos. O exame de Identificação Veicular realizado por peritos é uma das grandes demandas da perícia. É comum que, ainda hoje, apesar da popularização dos dispositivos móveis, os peritos façam uso de meios manuscritos para anotações. Além disso, a digitalização de processos do nosso cotidiano é uma realidade há alguns anos. O sistema desenvolvido visa otimizar o processo manual utilizado atualmente pelos peritos no momento do exame de Identificação Veicular. O desenvolvimento de solução para esse problema consiste em uma aplicação móvel para coleta de dados, uma aplicação web para geração do laudo e uma API do lado servidor que visam substituir o processo atual, buscando otimizar o tempo de trabalho e evitar problemas decorrentes de perda de informações. Por fim, pretende-se facilitar a geração de laudos a partir dos dados digitalmente coletados.

Palavras-chave: laudos periciais; identificação veicular; perito; dispositivo móveis.

ABSTRACT

The Criminal Forensics, an area of great importance for the judicial process, is charged with carrying out examinations to clarify criminal facts. The Vehicle Identification exam, performed by forensic experts, is one of the great demands in criminalistics. Using handwritten means by forensic experts still common for taking notes, despite the popularization of mobile devices. In addition, digitalizing daily processes has been a reality for years in most areas. Therefore, the present work proposes to develop a system for data collection during Vehicle Identification exams. This way, seeking to optimize the manual process currently used by forensics experts. The proposed solution consists in a mobile application for data collection, a web application capable of generating reports, and a server-side API that united aim to replace current handwritten process, optimize working time and avoid problems resulting from loss of information. Finally, it is intended to facilitate the generation of reports from digitally collected data.

Keywords: report; criminal forensics; vehicle identification; criminalistics; mobile devices.

LISTA DE QUADROS

Quadro 1 – Comparativos entre os Sistemas.	21
Quadro 2 – Requisitos Funcionais.	31
Quadro 3 – Requisitos Não-Funcionais.	31

LISTA DE ABREVIATURAS E SIGLAS

Siglas

ACC	Assistente de Cena de Crime
API	Interface de programação de aplicativo, do inglês <i>Application Programming Interface</i>
IDE	Ambiente de desenvolvimento integrado, do inglês <i>Integrated Development Environment</i>
GLB	Gerenciamento de Laudos Balístico
MVC	Modelo, Visão e Controle, do inglês <i>Model-View-Controller</i>
MVP	Mínimo Produto Viável, do inglês <i>Minimum Viable Product</i>
SDK	kit de desenvolvimento de software , do inglês <i>Software Development Kit</i>
SSP	Secretaria de Segurança Pública
TCC	Trabalho de Conclusão de Curso
RF	Requisito Funcional
RNF	Requisito Não Funcional
RENAVAM	Registro Nacional de Veículos Automotores
REP	Requisição de Exames Periciais
VM	Máquina Virtual, do inglês <i>Virtual Machine</i>

SUMÁRIO

1	INTRODUÇÃO	13
2	OBJETIVOS	15
2.1	OBJETIVO GERAL	15
2.2	OBJETIVOS ESPECÍFICOS	15
2.3	DIFERENCIAL TECNOLÓGICO	15
3	REVISÃO BIBLIOGRÁFICA	16
3.1	TRABALHOS RELACIONADOS	16
3.1.1	GLB - GERENCIAMENTO DE LAUDOS BALÍSTICOS	16
3.1.2	ACC - APLICATIVO ASSISTENTE DE CENA DE CRIME	16
3.1.3	VISTORIA DIGITAL 2	17
3.1.4	E-FORM	19
3.1.5	ELAUDO	19
3.2	REFERENCIAL TEÓRICO	22
3.2.1	REACT JS	22
3.2.2	REACT NATIVE	22
3.2.3	JAVASCRIPT	22
3.2.4	EXPRESS	23
3.2.5	SWAGGER	23
3.2.6	METODOLOGIA DE DESENVOLVIMENTO ÀGIL SCRUM	23
3.2.7	TESTES	23
4	METODOLOGIA	25
5	RESULTADOS ALCANÇADOS	27
5.1	LEVANTAMENTO E ANÁLISE DE REQUISITOS	27
5.1.1	MAPA CONCEITUAL	27
5.1.2	ARQUITETURA DO SISTEMA	29
5.1.3	DIAGRAMA DE CASO DE USO	30
5.1.4	REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS	30
5.2	MODELAGEM E DESENVOLVIMENTOS DE PROTÓTIPOS DE TELAS	31
5.2.1	MODELAGEM DO BANCO DE DADOS	32
5.2.2	PROTOTIPAÇÃO DAS TELAS	33

5.3	DESENVOLVIMENTO DA APLICAÇÃO WEB (SERVIDOR)	33
5.3.1	API/BACK-END	33
5.3.2	INTERFACE WEB	35
5.3.3	CRIAÇÃO DO BANCO DE DADOS	38
5.3.4	TESTES AUTOMATIZADOS	39
5.4	DESENVOLVIMENTO DO APLICATIVO MÓVEL	41
5.4.1	AMBIENTE DE DESENVOLVIMENTO	41
5.4.2	CADASTRO DE LAUDOS	41
5.4.3	ENVIO DO LAUDO	43
5.4.4	EDIÇÃO E EXCLUSÃO DO LAUDO	43
5.4.5	DEPLOY	44
6	CONSIDERAÇÕES FINAIS	45
6.1	TRABALHOS FUTUROS	45
	REFERÊNCIAS	47
	APÊNDICE A DIAGRAMA ENTIDADE-RELACIONAMENTO	49
	APÊNDICE B DOCUMENTO WORD - LAUDO	52

1 INTRODUÇÃO

No Paraná, segundo estatísticas oficiais da Secretaria de Segurança Pública (SSP, 2019), houveram em média 10.893 ocorrências de furto ou roubo de veículos no primeiro semestre de 2019. As estatísticas também apontam uma taxa de recuperação de 61,6% dos veículos, ou seja, em torno de 6.713 veículos furtados ou roubados foram recuperados. Estes veículos, na sua maioria, podem sofrer algum tipo de modificação em um de seus sinais identificadores, como chassi e motor. Quando necessário, cabe às autoridades policiais, judiciais ou do Ministério público solicitar ao órgão de perícia criminal um exame de Identificação Veicular a fim de testar a integridade, ou não dos mesmos.

A Perícia Criminal, área de suma importância para um processo judicial, é função do Estado e tem o encargo de realizar exames com objetivo de esclarecer fatos delituosos, desde avaliação de materiais até a elucidação de dinâmica criminosa, através da observação e análise de vestígios (PRADO, 2014). O Instituto de Criminalística é um órgão técnico-científico, responsável por realizar perícias relativas à criminalística, que tem o objetivo de auxiliar a justiça fornecendo provas técnicas. Através de seções, o Instituto de Criminalística está presente em dez cidades no Paraná: Curitiba, Guarapuava, Cascavel, Paranaguá, Foz do Iguaçu, Londrina, Umuarama, Ponta Grossa, Maringá e Francisco Beltrão (CRIMINALISTICA, 2019).

Policiais e investigadores, com a função de perito, são profissionais designados para realizar exames, por exemplo, de Identificação Veicular, e posteriormente devem transcrever os dados e as informações coletadas para documentos chamados Laudos Periciais. Os dados coletados por estes profissionais são de suma importância para averiguações subsequentes, além destes, os efeitos jurídicos também são baseados nas observações e conclusões feitas pelo perito.

O Laudo Pericial é uma prova fundamental em uma investigação, e tem como objetivo relatar a avaliação de um especialista para esclarecer uma determinada situação. Este documento deve ser claro e objetivo, fundamentado nos conhecimentos e técnicas do perito (PRADO, 2014).

De acordo com o Instituto de Criminalística do Paraná (CRIMINALISTICA, 2019), uma das grandes demandas de perícia é o exame de Identificação Veicular, que procura certificar a integridade de um veículo, ou seja, que não há adulteração de chassi, motor, placas e vidros, seja para averiguar sua identidade e/ou alterações não permitidas pelas leis vigentes.

A solicitação de perícia de Identificação Veicular é normalmente feita por ofício que parte do Ministério Público, delegacia, ou outros órgãos. O ofício é encaminhado ao Instituto de Criminalística com informações do veículo e a sua localização. Com isso, o responsável da seção designa um perito para realizar o exame, que então se desloca até local em que se encontra o veículo, geralmente em uma delegacia, realiza o procedimento de identificação veicular através de técnicas, reações químicas e cruzamento de dados com as montadoras dos veículos. As-

sim, o perito consegue determinar a verdadeira origem do veículo periciado, materializando seu trabalho no Laudo Pericial.

Não é incomum, ainda hoje, apesar da imensa popularização dos dispositivos móveis, que os profissionais usem meios manuscritos para realizar anotações, gerenciar processos e guardar informações. A coleta de dados, em diversas áreas, ainda é realizada de forma manuscrita, através de formulários em papel. Desta forma, o processo está sujeito a erros de preenchimento e leitura. Conseqüentemente, aumentam-se as chances de inserções equivocadas durante a transcrição dos dados coletados manualmente para um *software*. Tarefas como estas, que demandam muitas vezes tempo, são conduzidas dessa forma por não terem acesso a soluções digitais que satisfaçam suas necessidades, ou até mesmo pelo fato de tais soluções não existirem no mercado. Nesse sentido, existe um nicho a ser explorado no desenvolvimento de soluções digitais.

Na área de perícia, ter um sistema digital, que ofereça recursos ou funcionalidades, capaz de coletar, administrar e auxiliar na criação e visualização de dados, pode agilizar o processo de elaboração de laudos, otimizando o tempo de trabalho. Neste contexto, aproveitando as tecnologias existentes, a mobilidade e a conectividade disponíveis nos dispositivos móveis, o presente trabalho propõe o desenvolvimento de uma aplicação móvel que permita a coleta de dados de forma fácil, rápida e contextualizada para a geração de Laudos Periciais.

Almeja-se, com o aplicativo proposto, otimizar, no momento do exame, o cadastro de informações, como tipo de veículo, placa, marca, permitir ainda a adição de imagens e, por fim, permitir a geração do Laudo Pericial com as informações cadastradas. Ressalta-se que, no local da realização do exame (coleta de informações), não existe garantia de acesso à Internet, portanto o aplicativo deve prever essa situação. Ademais, destaca-se que os Laudos Periciais de veículos se dividem em quatro categorias: veículos, motocicletas, caminhões e semi-reboque. No escopo do presente Trabalho de Conclusão de Curso, será abordada uma aplicação voltada apenas para laudos de motocicletas.

2 OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é o desenvolvimento de um sistema para a coleta de dados no momento da realização do exame de Identificação Veicular e a geração do Laudo Pericial com estes dados.

2.2 OBJETIVOS ESPECÍFICOS

- Conhecer a rotina de trabalho de um perito durante um exame de Identificação Veicular;
- Desenvolver do aplicativo móvel para a coleta de dados;
- Desenvolver da aplicação web (servidor) para geração de laudos;
- Realizar experimentos na plataforma móvel.

2.3 DIFERENCIAL TECNOLÓGICO

O sistema proposto será de fácil uso e entendimento, funcionará sem acesso à Internet, e permitirá cadastro de informações em qualquer lugar onde o perito se encontre. Diante dessa possibilidade, o aplicativo irá substituir todo o processo manual, eliminando as anotações em papel atualmente utilizado pelo perito no momento da realização do exame. Além disso, destaca-se a capacidade de gerar laudos com os dados coletados.

3 REVISÃO BIBLIOGRÁFICA

A confecção de laudos veiculares é um problema específico na área de perícia técnica e, até o momento, não foram encontradas soluções que atendam as necessidades dos peritos no exame de Identificação Veicular. A presente Revisão Bibliográfica divide-se em dois tópicos: a apresentação de Trabalhos Relacionados (Seção 3.1); e a apresentação de um Referencial Teórico (Seção 3.2) contemplando conceitos e tecnologias que fundamentam a solução proposta.

3.1 TRABALHOS RELACIONADOS

3.1.1 GLB - GERENCIAMENTO DE LAUDOS BALÍSTICOS

Mognon (2019), em seu Trabalho de Conclusão de Curso (TCC), propôs um sistema para a confecção de laudos balísticos. O Gerenciamento de Laudos Balísticos (GLB) é um sistema gerador de laudos balísticos que permite a confecção de documentos com eficiência e estabilidade através de preenchimentos de formulários. Além disso, dentre as funcionalidades do GLB, é possível: cadastrar armas; adicionar ou remover informações específicas e imagens; visualizar os dados cadastrados; gerar relatórios em formato PDF e DOC (Microsoft Word). Por se tratar de um sistema específico voltado apenas ao contexto de laudos balísticos, o GLB não trata outros tipos de laudos, como, por exemplo, os laudos veiculares. Além disso, o GLB não dispõe de uma aplicação móvel capaz de coletar dados em locais sem acesso à Internet.

As figuras 1 e 2 apresentam exemplos de telas do sistema GLB. Essas telas representam o cadastro de uma arma e a visão geral do laudo. Sistema foi desenvolvido com base no padrão Model-View-Controller (MVC), utilizando a linguagem PHP em conjunto do *framework* Laravel.

3.1.2 ACC - APLICATIVO ASSISTENTE DE CENA DE CRIME

O aplicativo Assistente de Cena de Crime (ACC), apresentado por Amorim *et al.* (2019), foi desenvolvido para assistência policial na coleta de dados na cena de crime, permitindo o cadastro de vítimas, evidências, depoimentos e outros dados em tempo real. Evitando perda de dados e garantindo a segurança das informações colhidas no local, o aplicativo é capaz de gerar relatórios sobre a cena de crime, proporcionando rapidez na coleta e compartilhamento de dados.

Conforme expresso na Figura 3, a tela do aplicativo ACC permite ao usuário visualizar as ocorrências, filtrar por situação, intervalo de tempo e ainda gerar relatórios. Além disso, conforme expresso na Figura 4, o aplicativo ACC apresenta uma tela de formulário, em que o usuário informa a natureza do delito, local e data dos fatos, entre outras informações adicionais.

Figura 1 – Tela de Cadastro de Arma.
Fonte: Mognon (2019).

Figura 2 – Tela de Visão Geral do Laudo.
Fonte: Mognon (2019).

3.1.3 VISTORIA DIGITAL 2

O Departamento de Trânsito do Paraná DETRAN (2016) desenvolveu um aplicativo para vistoria digital de veículos, o Vistoria Digital 2. O aplicativo tem como principais funcionalidades cadastrar dados de identificação, observações e inclusão de fotos do veículo. A implementação contempla as plataformas Android e iOS, permite o funcionamento off-line e visa substituir o

uso de papel por arquivos eletrônicos. A Figura 5 ilustra a tela usada para cadastrar uma nova vistoria (esquerda), que permite ao usuário selecionar o automóvel, além de informações do proprietário e da placa. Na tela de finalizar a vistoria (direita), o usuário pode visualizar as imagens adicionadas e adicionar informações, como: itens reprovados, nº de lacre da placa ou observações.

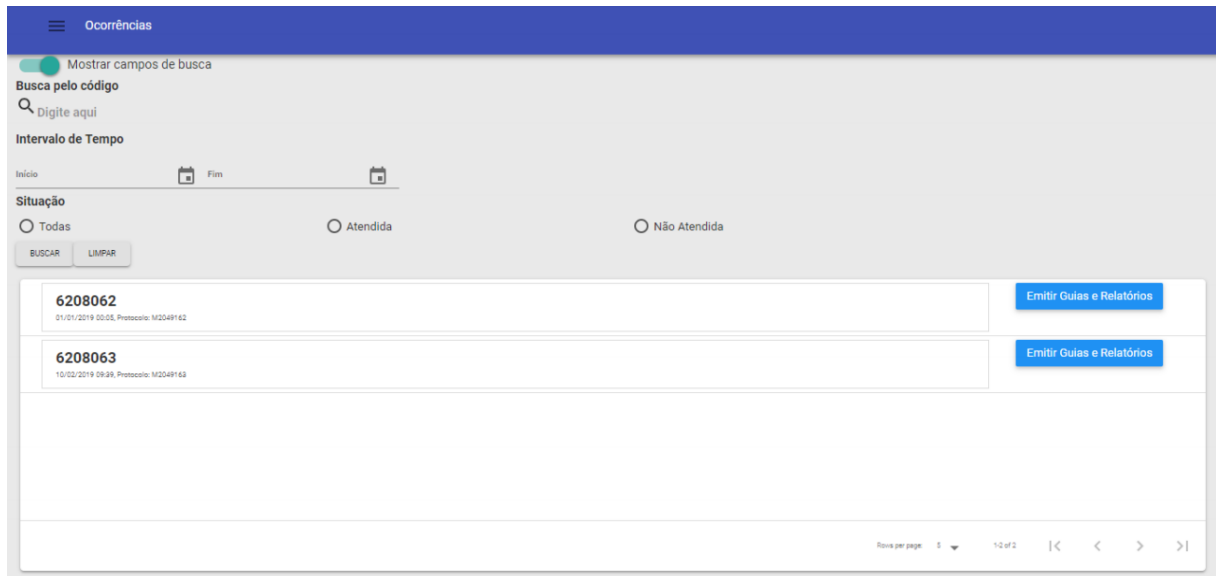
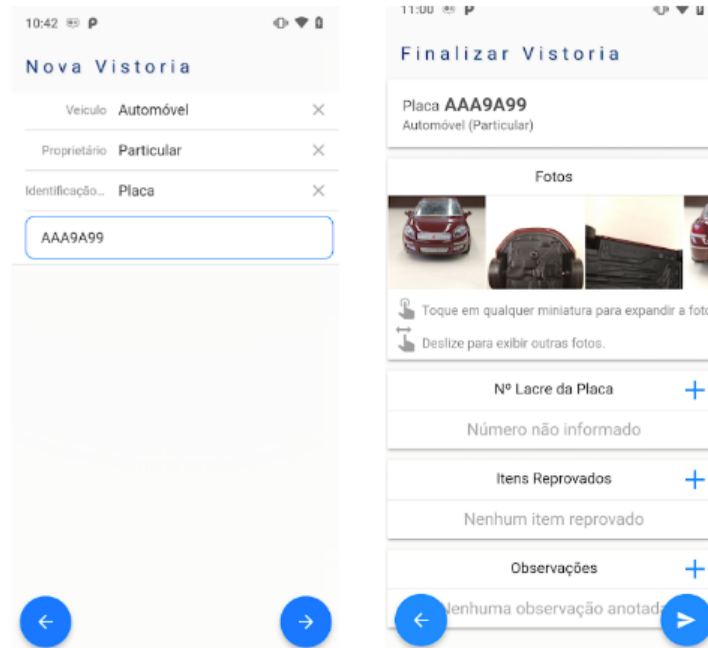


Figura 3 – Tela de Ocorrências.
Fonte: Amorim *et al.* (2019).

Figura 4 – Tela do Formulário.
Fonte: Amorim *et al.* (2019).

Além disso, o aplicativo Vistoria Digital 2, possui um formulário em que o usuário preenche informações como a placa, o número do RENAVAM, o chassi, o lacre, o motor, além de poder incluir fotos de cada item vistoriado, itens de segurança e até mesmo avarias encontradas. Como desvantagem, não é possível gerar relatórios, mas apenas o envio das informações para o DETRAN (2016).



**Figura 5 – Tela Cadastro de Vistoria e Tela de finalizar a vistoria.
Fonte: Loja de Aplicativos Google Play Store.**

3.1.4 E-FORM

De acordo com Azevedo, Miazaki e Porfirio (2016) o e-Form é um *software* para aplicação de questionários eletrônicos, que permite aos usuários criar e gerenciar questionários para coleta de dados por meio de *smartphones*. O *software* visa automatização de processos, otimização do tempo no preenchimento de questões e substituição de todo o processo manual, eliminando assim o papel e as possíveis falhas humanas. O sistema e-Form pode ser utilizado de maneira off-line, permitindo o usuário realizar entrevistas sem acesso à internet. Também oferece gerenciamento de gráficos para análise com base nos resultados do questionário. Por ser um *software* voltado para pesquisas, possui questões pré-definidas e não gera documentos a partir dos dados coletados.

3.1.5 ELAUDO

Desenvolvido por Canto (2017), o e-Laudo é um aplicativo gratuito para que consultores e produtores rurais possam registrar o acompanhamento de sua lavoura, permite tirar fotos, registrar observações, fazer recomendações, além de permitir a seleção de áreas inspecionadas direto no mapa. O aplicativo pode ser utilizado off-line, sem a necessidade de acesso à Internet, todavia, os registros podem também serem salvos na nuvem, permitindo a geração de um PDF com todas as informações e ainda permitindo o compartilhamento entre produtor, consultor, entre outros usuários. O aplicativo e-Laudo tem como objetivo auxiliar na elaboração de laudos de visitas técnicas, um documento gerado pelo consultor rural a cada visita realizada em uma

propriedade. Além disso, o aplicativo também visa a otimização do tempo, eliminando o papel e o retrabalho de passar informações para o computador, busca a padronização dos laudos, além de controle dos laudos emitidos e consulta os dados da últimas visitas. Por se tratar de um aplicativo voltado para área rural, não atende outros tipos de laudos. A Figura 6 apresenta a tela inicial do aplicativo e a tela de menu.



Figura 6 – Tela Inicial e Tela Menu de Opções do e-Laudo.
Fonte: Loja de Aplicativos Google Play Store.

Dentre os sistemas citados anteriormente, alguns recursos são destacados: o funcionamento off-line, o cadastro de imagens, o gerenciamento de documentos *Word* e PDF, e a interação através de formulários simples e objetivos. Estes recursos mostraram-se interessantes e foram considerados como base e fontes de inspiração para o desenvolvimento do sistema proposto no presente trabalho. Embora existam soluções similares, destacando-se o GLB, que mais se assemelha ao sistema proposto, a extensão dos sistemas prévios é desconsiderada devido à necessidade de uma interface de programação (API), restrições de conexão, necessidade de oferecer uma aplicação móvel, e restrição para tráfego de dados pela Internet¹. Assim, a especificação e implementação de um novo sistema é justificada.

¹ Restrição de segurança imposta pelo perito do Instituto de Criminalística.

3.2 REFERENCIAL TEÓRICO

A escolha das tecnologias se deve à popularização dos *smartphones* e à conectividade dos dispositivos móveis, além do poder de processamento, armazenamento, ambiente de utilização e também características específicas como sensores, câmera, bateria, entre outros. O desenvolvimento de aplicações móveis pode ser realizado por meio de ferramentas nativas (tal como a SDK exclusiva do Android ²), ou ainda com o uso de *frameworks* para a construção de aplicações multiplataforma, que favorecem a portabilidade de aplicações em trabalhos futuros, permitindo que um mesmo código-fonte possa ser compilado e exportado para Android, iOS e Web.

3.2.1 REACT JS

O React JS é uma ferramenta *open source* bastante popular, desenvolvida pelo Facebook, para a construção de interfaces em aplicações multiplataforma. Segundo o website oficial (REACT, 2020), React é uma biblioteca para construção de interfaces de usuário. React não é um *framework*, nem mesmo é exclusivo para web, podendo ser utilizado com outras bibliotecas, como exemplo React Native para construção de aplicativos móveis.

3.2.2 REACT NATIVE

O React Native, por sua vez, é uma estrutura JavaScript que possibilita o desenvolvimento aplicativos móveis nativos para iOS e Android. Ele é baseado no React, a biblioteca JavaScript do Facebook para criar interfaces de usuário, mas, em vez de segmentar o navegador, ele segmenta plataformas móveis. Além disso, a maior parte do código pode ser compartilhado entre as plataformas (EISENMAN, 2015). Recentemente, a plataforma Expo foi estabelecida visando unificar o desenvolvimento móvel e web, unindo as capacidades do React tradicional com as do React Native em um único ambiente de desenvolvimento (EXPO, 2020).

3.2.3 JAVASCRIPT

Tratando-se do ambiente de desenvolvimento, a linguagem de programação a ser utilizada é o JavaScript ou JS. Criado em 1995, e atualmente disponível na versão ES2021, o JavaScript é uma linguagem de programação interpretada e orientada a objetos. É também uma das linguagens mais utilizadas no mundo, embora seja conhecida como uma linguagem web, com o surgimento dos *frameworks* de desenvolvimento mobile multiplataforma sua utili-

² Portal de desenvolvedores do Android: <https://developer.android.com/>

zação tem aumentado cada vez mais. Além disso, é muito utilizada em ambientes externos ao navegador Web, como, por exemplo servidores Node.js (WIKIPEDIA, 2021).

3.2.4 EXPRESS

O NodeJS possui um framework chamado Express, onde é possível criar aplicações back-end. De acordo com (HAHN, 2016) o Express abstrai a complexidade do Node.js oferecendo funcionalidades significativas, como recursos para o desenvolvimento de APIs. Neste contexto, o framework fornece um conjunto de métodos utilitários HTTP para criar APIs de forma simples e rápida. EXPRESS (2017)

3.2.5 SWAGGER

Ao consumir uma API, torna-se necessário conhecer suas funcionalidades disponíveis e detalhes de utilização, como: recursos, métodos, Content-Types e outras informações. O Swagger é uma ferramenta utilizada para modelagem, documentação e geração de código para APIs REST, e permite a documentação e publicação dos serviços disponíveis. Dentre os recursos do Swagger, destacam-se ser uma ferramenta interativa e a possibilidade de visualizar e testar todos os métodos disponibilizados. SMARTBEAR (2020)

3.2.6 METODOLOGIA DE DESENVOLVIMENTO ÀGIL SCRUM

Tão importante quanto a definição das tecnologias é a escolha da metodologia de desenvolvimento. Segundo Pressman e Maxim (2016), o desenvolvimento ágil de *software* é uma abordagem que surgiu formalmente após a publicação do Manifesto Ágil por um grupo de especialistas da área. Os métodos ágeis focam os esforços na entrega contínua de *software* em curtos períodos de tempo, e têm como uma das principais características desenvolvimento iterativo e incremental.

3.2.7 TESTES

Diante do uso de metodologias ágeis, destaca-se a etapa de testes do software. Os testes têm como finalidade garantir que a integração de novas funcionalidades ao sistema não quebrem o funcionamento das já existentes. Tratando-se de testes no contexto do presente trabalho, o framework open source Jest, criado pelo Facebook, mostra-se uma opção relevante, podendo ser aplicado em Node, JavaScript puro, React, Angular, Vue e outros *frameworks* (FACEBOOK, 2017). O Jest inclusive já vem integrado ao ambiente de desenvolvimento do React

Native, não exigindo configuração, sendo necessário apenas mantê-lo como dependência do projeto.

4 METODOLOGIA

Este capítulo descreve, de forma sequencial, os passos elaborados para a execução do trabalho. Nos itens a seguir estão descritos os procedimentos metodológicos adotados.

- Levantamento e análise requisitos:
 - Identificar características, necessidades e funcionalidades desejadas;
 - Identificar requisitos funcionais;
 - Identificar requisitos não-funcionais;
 - Elaborar um mapa conceitual com as funcionalidades do sistema;
 - Especificar a arquitetura do sistema;
 - Elaborar um diagrama comportamental dos casos de uso;
- Modelar o banco de dados do sistema:
 - Representar os dados do mapa conceitual como tabelas;
 - Definir relacionamentos entre as tabelas;
 - Criar o diagrama Entidade-Relacionamento;
- Modelar e desenvolver protótipos de telas:
 - Definir o estilo visual (CSS) da aplicação mobile;
 - Tela de login;
 - Tela inicial;
 - Telas do formulário de cadastro;
 - Tela de listagem dos laudos;
- Definir as tecnologias e ferramentas para o desenvolvimento do sistema;
- Desenvolver o aplicativo móvel para a coleta de dados:
 - Criar página para login;
 - Criar página principal;
 - Criar página com formulário em etapas;
 - Criar página para visualizar a lista de laudos;
 - Recurso de armazenamento de imagens;
 - Recurso de criação de laudo;
 - Recurso de envio de laudo;

- Recurso de exclusão de laudo;
- Recurso de edição de laudo;
- Recurso para verificar conexão com a Internet;
- Desenvolver a aplicação web (servidor) para geração do laudo:
 - Criar API para registro e gerenciamento de laudos:
 - * Recurso de criação de laudo;
 - * Recurso de edição de informações de laudo;
 - * Recurso de exclusão de laudo;
 - * Recurso para gerar laudo em Docx;
 - * Recurso de autenticação de usuários;
 - * Criar testes automatizados para as principais funcionalidades;
 - * Configuração servidor;
 - * Deploy;
 - Criar aplicação front-end para gerar documento:
 - * Criar página para login;
 - * Criar página para lista de laudos;
 - * Recurso para obter o download de um laudo;
 - * Recurso para a exclusão de um laudo.

5 RESULTADOS ALCANÇADOS

Nesta seção são apresentados os resultados alcançados com a execução das etapas metodológicas descritas no Capítulo 4: levantamento e análise requisitos; modelagem e desenvolvimento de protótipos de telas; desenvolvimento do aplicativo; desenvolvimento da aplicação web; e teste de usabilidade.

5.1 LEVANTAMENTO E ANÁLISE DE REQUISITOS

A primeira etapa de desenvolvimento do projeto abordou o levantamento dos requisitos. Foi utilizada a proposta de descrição de requisitos classificada por Sommerville como Requisitos Funcionais e Não-Funcionais (SOMMERVILLE, 2008). Para melhor entendimento do problema, foi apresentado um modelo do Laudo Pericial confeccionado pelo usuário e elaborado um mapa conceitual do sistema (Seção 5.1.1). A partir do mapa conceitual, foram definidas as funcionalidades do sistema, priorizando-se as mais relevantes, dando origem à arquitetura exposta na Seção 5.1.2. A seguir, a listagem dos requisitos do sistema foi dividida entre Requisitos Funcionais (RF) e Não-Funcionais (RNF), expostos na Seção 5.1.4. Em seguida, as principais funcionalidades do sistema foram detalhadas em um diagrama de Casos de Uso, exposto na Seção 5.1.3.

5.1.1 MAPA CONCEITUAL

A fim de melhor compreender o contexto do problema e os detalhes das informações que compõe um Laudo Pericial, foi elaborado um modelo de Laudo Pericial e seguindo como modelo de laudo na Figura 7, as informações que devem ser salvas no aplicativo são representadas em negrito no documento.

Em seguida, foi elaborado um mapa conceitual, exposto na Figura 8, que detalha os padrões institucionais adotados no Instituto de Criminalística durante a confecção de laudos veiculares. Os nodos apresentados no mapa são atributos detalhados de cada veículo¹.

Um laudo é composto por um cabeçalho, que deve conter: o número da Requisição de Exames Periciais (REP), número do ofício, nome do diretor do órgão, perito designado, natureza do exame, órgão solicitante, e data de solicitação. Em seguida, o laudo contém as informações do veículo, que se dividem em quatro categorias: veículos, motocicletas, caminhões e semi-reboque. Motocicletas contém apenas chassi e motor. Já veículos, como carros, possuem agregados, peças como vidros e etiquetas. Além disso, todos os tipos de veículos contém dados básicos, como marca, modelo, ano de fabricação, placa, etc. O laudo também contempla o exame das peças, em que são descritas detalhadamente informações que visam esclarecer

¹ O mapa completo, com todos os nodos abertos, encontra-se disponível em: <https://bit.ly/3fem862>

se há os sinais de adulteração nas numerações identificadoras que impactam no resultado do exame. Por fim, o documento contém um texto padrão contendo o nome do perito que realizou o laudo.

FLS.1 laudo(Nº DO LAUDO)
LAUDO DE EXAME DE (TIPO DE EXAME)
<p>Aos DATA POR EXTENSO, nesta cidade de CIDADE DA SEÇÃO e no INSTITUTO DE CRIMINALÍSTICA do Estado, foi designado pelo Diretor do Instituto NOME DO DIRETOR por indicação do chefe da Seção, o Perito Criminal NOME DO PERITO, para proceder ao exame no veículo mencionado no Ofício nº. Nº DO OFÍCIO, NATUREZA DO EXAME recebido dia DATA DA SOLICITAÇÃO, oriundo ÓRGÃO SOLICITANTE.</p> <p>Em consequência, o Perito procedeu ao exame solicitado, relatando-o com a verdade e com todas as circunstâncias relevante, da forma como segue:</p> <p>MOTIVO DA PERÍCIA</p> <p>Depreende-se da leitura do ofício supracitado, que a perícia tem por finalidade proceder ao exame nas enumerações do veículo acima mencionado.</p> <p>DO EXAME</p> <p>O veículo apresentado ao exame foi periciado no depósito de veículos da delegacia solicitante. Trata-se de um automóvel de marca de fabricação MARCA de cor COR, ostentando placas de licenciamento PLACA (CIDADE).</p> <p style="text-align: center;">IMAGEM DO VEÍCULO</p> <p>I. NUMERAÇÃO DO CHASSI</p> <p>Ao exame do suporte, onde se acha gravada a numeração de chassi após a devida limpeza, verificou o perito sinais de operação ali procedida, que consistiu no TIPO DE ADULTERAÇÃO, por ação abrasiva, da superfície, acarretando a destruição da numeração original e possibilitando a gravação da atual NUMERAÇÃO DO CHASSI adulterada.</p> <p>Submetida a superfície acima referida a tratamento químico metalográfico, destinado a revelar a numeração original destruída obteve-se a identificação dos caracteres NUMERAÇÃO DO CHASSI REVELADO, sendo estes a numeração original do chassi desse veículo.</p> <p style="text-align: center;">IMAGEM DA NUMERAÇÃO DO CHASSI</p> <p>Este laudo foi redigido pelo Perito NOME DO PERITO e disponibilizado em arquivo digital contendo uma folha de rosto, QUANTIDADE DE PAGINAS, pagina(s). Por nada mais haver e sendo essas declarações que tem a fazer, deu por findo o exame solicitado que de tudo se lavrou o presente laudo, o qual segue digitalmente assinado.</p>

Figura 7 – Modelo de Laudo Pericial.
Fonte: Elaborado pela autora.

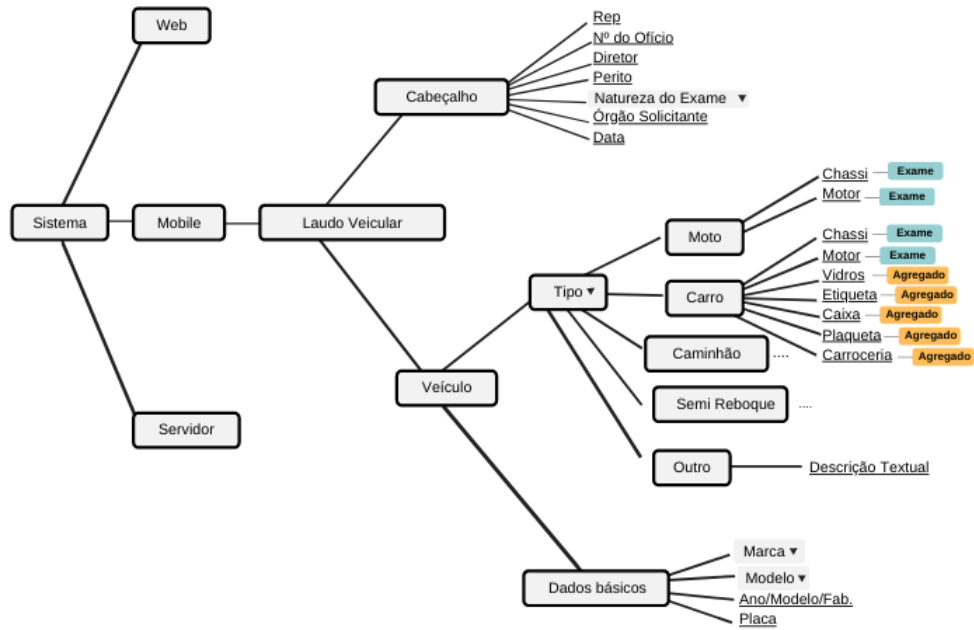


Figura 8 – Mapa Conceitual.
Fonte: Elaborado pela autora.

5.1.2 ARQUITETURA DO SISTEMA

Na Figura 9 é possível visualizar o modelo de arquitetura do sistema e a forma com que o Servidor Web se integra com as ferramentas, disponibilizando acesso ao dispositivo por meio de uma interface de programação de aplicativo (API), além de disponibilizar a interface Web para acesso via Intranet. O sistema desenvolvido deve permitir a coleta de dados em qualquer ambiente por meio de um dispositivo móvel. Utilizando este dispositivo, os peritos efetuam a coleta das informações e enviam os dados para o servidor, essa comunicação só deve ocorrer via Intranet quando o perito estiver na unidade, evitando o tráfego de dados pela Internet (requisito de segurança estabelecido pelo Instituto de Criminalística). O aplicativo móvel permite armazenar, consultar e editar informações relativas aos veículos antes de fazer o envio para o servidor. O aplicativo móvel tem suas funções reduzidas, se comparada à aplicação Web, pois o seu principal objetivo é a coleta de informações no momento de realização do exame, enquanto a interface Web deve fornecer recursos para a geração dos laudos.

Ainda considerando a Figura 9 destaca-se o uso das tecnologias mencionadas no Referencial Teórico (Seção 3.2): Node.JS, JavaScript, SQLite (banco para testes no ambiente de desenvolvimento), empregadas no **lado servidor**; React.JS, React Native, JavaScript, SQLite (embarcado na aplicação móvel), empregadas no **lado do cliente**. Ademais, a metodologia ágil Scrum é empregada em ambas as etapas de desenvolvimento.

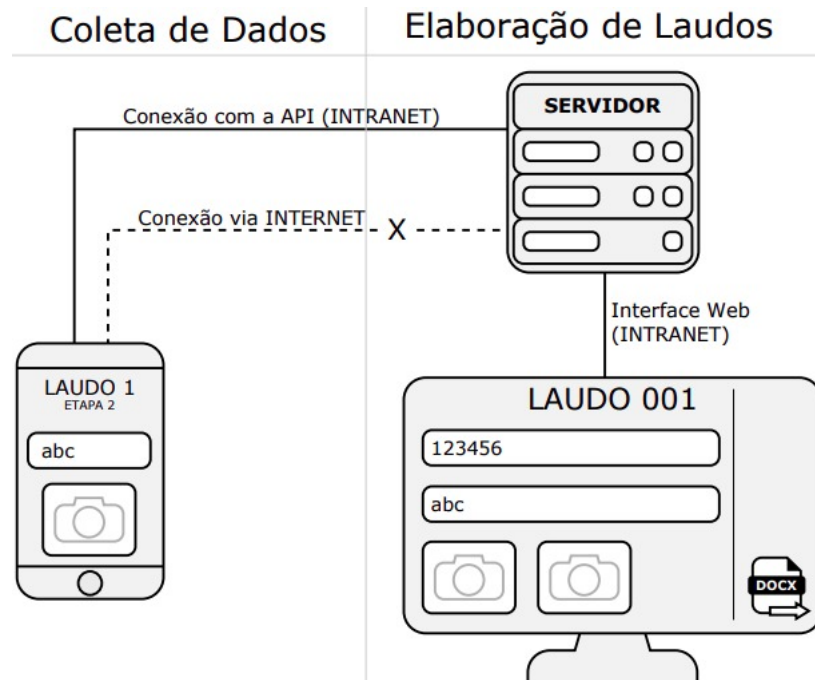


Figura 9 – Arquitetura do Sistema.
Fonte: Elaborado pela autora.

5.1.3 DIAGRAMA DE CASO DE USO

De acordo com (SOMMERVILLE, 2008), o caso de uso tem o objetivo de identificar as interações que podem ocorrer no sistema. Essas interações podem ser apresentadas através de diagramas que visam auxiliar na elaboração e entendimento dos requisitos. A Figura 10 expõe os casos de uso definidos para o sistema, onde é possível visualizar as principais atividades do usuário. Inicialmente o usuário poderá fazer login. Após o login ele será dirigido à tela de cadastro, onde poderá cadastrar as informações dos exames e adicionar imagens. Além disso, o usuário poderá enviar os dados coletados e por fim, solicitar a geração do laudo e efetuar o download do mesmo.

5.1.4 REQUISITOS FUNCIONAIS E NÃO-FUNCIONAIS

O Quadro 2 apresenta os Requisitos Funcionais, sua rastreabilidade, ou seja, vinculação com os casos de uso associados, e classificação de acordo com seu nível de prioridade. Em adição, os requisitos Não-Funcionais são listados no Quadro 3.

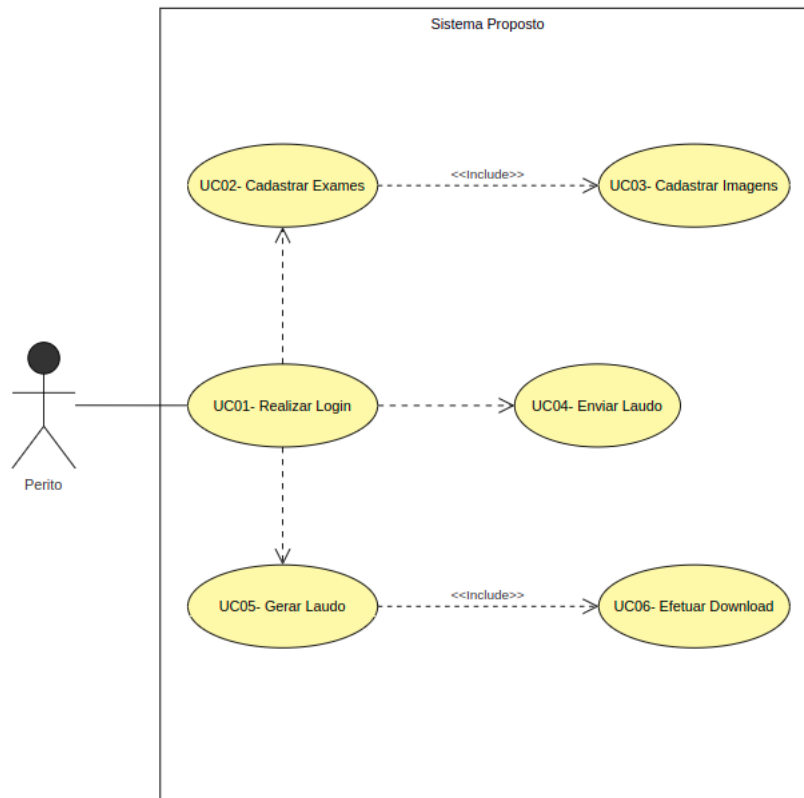


Figura 10 – Diagrama de Casos de Uso.

Fonte: Elaborado pela autora.

Quadro 2 – Requisitos Funcionais.

Nº	Requisitos Funcionais	Caso de Uso	Prioridade
[RF01]	O sistema deve prover autenticação de usuários para acesso ao aplicativo.	UC01	Média
[RF02]	O sistema deve permitir ao usuário cadastrar os dados coletados no Exame de Identificação Veicular.	UC02	Alta
[RF03]	O sistema deve permitir ao usuário o cadastro de imagens dos veículos.	UC03	Alta
[RF04]	O sistema deve enviar os dados coletados para o servidor.	UC04	Alta
[RF05]	O sistema deve permitir ao usuário gerar laudo.	UC05	Alta
[RF06]	O sistema deve permitir ao usuário efetuar o download do laudo apenas na aplicação web.	UC06	Média

Quadro 3 – Requisitos Não-Funcionais.

Nº	Requisitos Não-Funcionais	Prioridade
[RF01]	Manter compatibilidade com os principais navegadores.	Alta
[RF02]	Utilizar um padrão de codificação a fim de facilitar manutenções ou trabalhos futuros.	Alta
[RF03]	A interface do sistema deverá ser agradável e objetiva, funcionalidades e informações deve ser intuitivas.	Alta

5.2 MODELAGEM E DESENVOLVIMENTOS DE PROTÓTIPOS DE TELAS

Após o levantamento de requisitos, são apresentados os resultados da segunda etapa metodológica: modelagem e desenvolvimento de protótipos. Segundo Pressman e Maxim

(2016), o modelo de dados descreve estrutura lógica processada pelo sistema. O resultado da modelagem é exposto na Seção 5.2.1. Em seguida, na Seção 5.2.2, são apresentados os modelos de protótipos das telas da aplicação web e do aplicativo móvel.

5.2.1 MODELAGEM DO BANCO DE DADOS

A estrutura do banco de dados foi representada por um Diagrama Entidade-Relacionamento, detalhado no Apêndice A. O diagrama expressa as tabelas definidas para o armazenamento de dados dos veículos, suas peças, e informações gerais dos laudos. Além disso, o diagrama mostra as entidades, seus atributos e os relacionamentos na especificação do sistema.

Dentre as tabelas, é conveniente esclarecer a funções e os relacionamentos das mais importantes para o sistema:

- **report:** tabela em que serão armazenados as informações gerais referentes aos laudos. Atributos mais importantes: *id* (chave primária), *rep* (nº de registros de requisições de exames Periciais), e *user* (perito responsável pelo laudo);
- **vehicles:** tabela responsável por armazenar os dados básicos de cada veículo, como placa, ano, modelo de fabricação e estado de conservação. Está relacionada aos tipos de veículos existentes para exame, como motocicleta, carro, caminhão, semi-reboque. Esta tabela também armazena informações referentes às condições do veículo;
- **exams:** tabela mais importante do sistema, onde serão armazenados os dados de cada exame, sempre que houver uma nova peça a ser examinada, esta tabela será alterada;
- **motorcycles:** um dos tipos de veículos, responsável relacionar o laudo ao veículo. Esta tabela possui um relacionamento 1 para N com a tabela de exames (*exams*), pois cada veículo pode possuir vários exames;
- **aggregates:** tabela onde serão armazenados os agregados, que são informações adicionais como vidro, eixos, plaquetas. Dentre seus atributos, os mais importantes são: *id* da peça, tipo de gravação e o exame qual está relacionada;
- **pieces:** tabela responsável por armazenar o nome da peça a qual será feito o exame, possui um campo chamado *glasses* caso seja vidro.

5.2.2 PROTOTIPAÇÃO DAS TELAS

Esta etapa objetiva apresentar o design para o sistema. A Figura 11 ilustra a tela de cadastro dos dados básicos do veículo (esquerda). Esta tela também permite ao usuário cadastrar novas marcas de carros caso alguma não exista previamente no banco de dados. Na tela de cadastro da identificação veicular (direita), o usuário deve cadastrar a numeração identificadora revelada no exame.

The image displays two side-by-side mobile application screens. Both screens feature a blue header with a menu icon on the left and a police badge logo on the right. A progress indicator at the top of each screen shows four steps: 'Informações Gerais', 'Dados do Básicos', 'Peça', and 'Exame', with the current step highlighted by a checkmark.

Etapa 2 - Dados básicos do Veículo (Left Screen): This screen contains several input fields: 'Placa *', 'Marca' (with a dropdown arrow and a '+ Cadastrar' button), 'Modelo', 'Ano/Modelo/Fab.', and 'Cor' (with a dropdown arrow). At the bottom, there are three radio buttons for 'Estado de Conservação': 'Bom' (selected), 'Regular', and 'Mau'. Navigation buttons 'Voltar' and 'Próximo' are located at the bottom.

Etapa 4 - Exame (Right Screen): This screen shows a 'Peça: Chassi' field, followed by two checkboxes: 'Íntegro' (checked) and 'Adulterado'. Below this is a section titled 'Insira a numeração' with a grid of input fields for characters: J, N, 3, M, S, 3, 7, A, 9, P, W, 2, 0, 2, 9, 2, 9. Navigation buttons 'Voltar' and 'Próximo' are at the bottom.

Figura 11 – Tela Cadastro dos Dados do veículo e Tela de Cadastro da Identificação Veicular.
Fonte: Elaborado pela autora.

A Figura 12 representa a tela para listagem dos dados Coletados que aguardam a conclusão e envio para o servidor ².

5.3 DESENVOLVIMENTO DA APLICAÇÃO WEB (SERVIDOR)

5.3.1 API/BACK-END

O armazenamento dos dados coletados no lado servidor (*back-end*) exigiu o desenvolvimento de uma API, que serve como camada intermediária para permitir a comunicação entre aplicativo móvel e os sistemas de persistência e processamento remotos. Na estrutura adotada, o aplicativo realiza envios de dados através de uma conexão HTTP, os dados são convertidos em uma estrutura denominada JSON (*JavaScript Object Notation*).

² Todos os protótipos de tela podem ser encontrados online em: <https://bit.ly/3aJROCa>.



Figura 12 – Tela de Listagem Cadastros Dados Coletados.
Fonte: Elaborado pela autora.

A API foi desenvolvida utilizando a plataforma Node.js, juntamente com o framework Express.js. Os laudos devem ser coletados pelo aplicativo móvel, e então enviados por completo para o servidor, assim todas as informações referentes ao laudo são transmitidas em uma única requisição. Inicialmente toda a implementação do mecanismo de armazenamento dos laudos foi feita em um único modelo chamado *Report*, que era responsável por inserir os laudos no banco. Durante o desenvolvimento percebeu-se que o modelo *Report* estava acumulando muitas responsabilidades, e conseqüentemente dificultando a implementação de validações e a manutenção do código. Para facilitar a implementação e possibilitar a expansão da API para atender vários outros tipos de veículos, ou até mesmo vários tipos de laudos, foi implementado o Padrão de Projeto Factory (FREEMAN *et al.*, 2009). Com isso, as responsabilidades foram divididas em tarefas menores e independentes, encapsuladas em diferentes classes, e gerenciadas pelo objeto Factory.

Para cadastrar um novo laudo o usuário deve fazer uma requisição POST para a rota `/v1/reports` com as informações do Laudo, incluindo as imagens do veículo em base64, que são posteriormente armazenadas como arquivos. A forma como o JSON deve ser submetido para a API é especificado na documentação Swagger da API, como exemplificado na Figura 13.

Após o cadastro de um laudo, a API também permite gerar um documento em formato Docx. Para isso foram realizados testes com algumas bibliotecas, a que mais se aproximou do formato e padrão utilizado pelo perito foi a biblioteca Docx³ que fornece a capacidade de gerar e

³ Docx, website oficial: <https://docx.js.org>

gerenciar arquivos *Word Docx* dentro da própria aplicação com facilidade e sem dependências externas. Uma vez criado o documento, é possível modificá-lo e adicionar as informações de forma dinâmica.

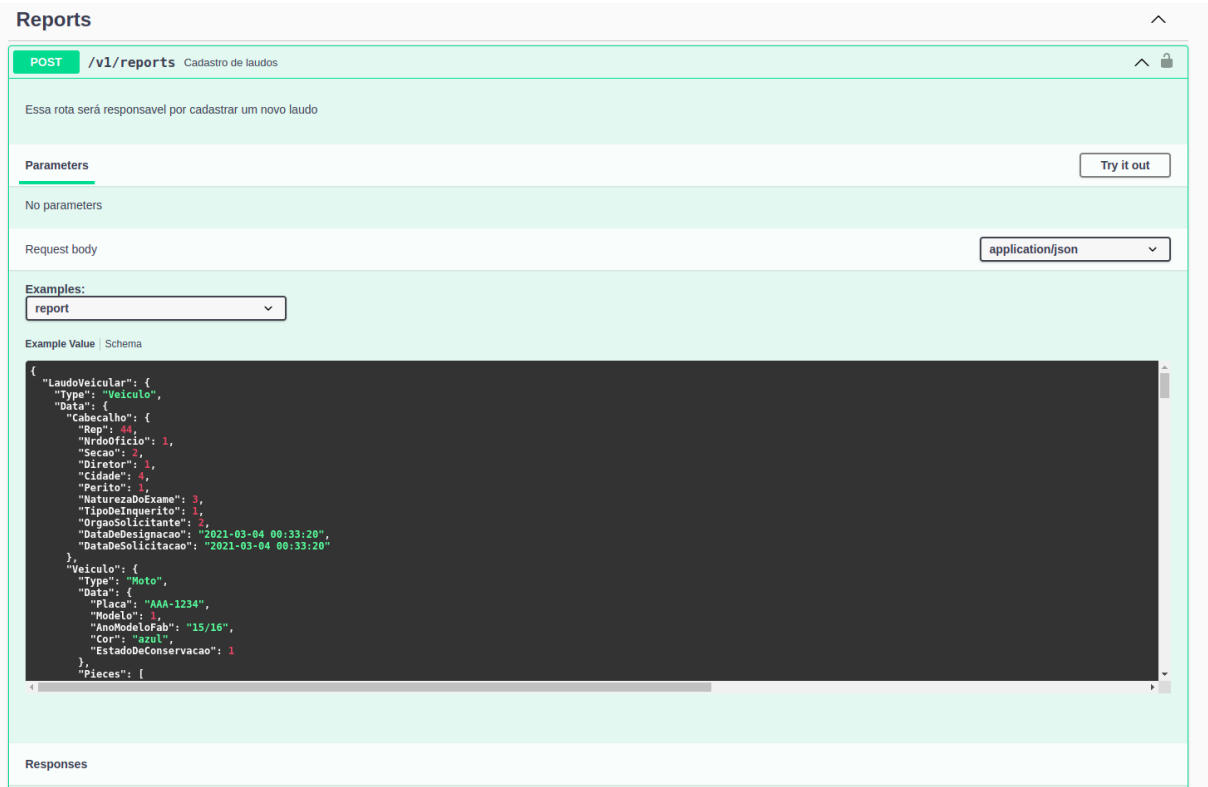


Figura 13 – Exemplo Requisição POST
Fonte: Elaborado pela autora.

A principal diferença de um laudo tradicional para um laudo de exame veicular é que para cada peça examinada o texto deve se adequar conforme as informações. A Figura 14 apresenta um trecho do resultado do documento gerado para uma peça íntegra. O resultado completo de um laudo íntegro gerado pela API encontram-se no Apêndice B.

Para gerar o documento, o usuário deverá realizar uma requisição GET para rota `/v1/reports/doc/:reportId` sendo `:reportId` a forma de identificar uma variável na URL que é substituída pelo valor do identificador do laudo. Além disso, a resposta da requisição retornará um arquivo zip contendo o Documento Word e todas as imagens relacionadas ao laudo.

5.3.2 INTERFACE WEB

A interface da aplicação Web foi desenvolvida com React JS. Visando mais simplicidade e segurança com os dados, os usuários são inseridos no banco de dados apenas por meio do back-end, sendo esta responsabilidade exclusiva dos administradores do sistema.

A Figura 15 apresenta a tela de login da aplicação web, o usuário deverá inserir suas credenciais nos campos e depois clicar no botão entrar.

O referido veículo apresenta a seguinte numeração identificadora: **2AEHD5RPA0KHK3894**. E número do motor **2AEHD5RPA0KHK3432**. Tendo-se procedido ao acurado exame nas referidas numerações, bem como nas superfícies que lhe servem de suporte, **não foram observados quaisquer sinais ou vestígios de adulteração.**



NUMERAÇÃO DO CHASSI



NUMERAÇÃO DO MOTOR

Este laudo foi redigido pelo Perito **José Roberto** e disponibilizado em arquivo digital contendo uma folha de rosto dois e disponibilizado em arquivo digital contendo uma folha de rosto e mais uma página. Por nada mais haver e sendo essas as declarações que tem a fazer, deu-se por findo o exame solicitado que de tudo se lavrou o presente laudo, o qual segue digitalmente assinado.

Figura 14 – Exemplo do Docx gerado com peças íntegras.

Fonte: Elaborado pela autora.

Ao acessar a tela principal exposta na Figura 16 são exibidos os laudos cadastrados pelo usuário com os seus respectivos dados, sendo eles: *rep*, número do ofício, cidade, órgão solicitante. Além disso, é possível fazer uma pesquisa pela *rep* do laudo.

Na tela principal, o usuário tem algumas opções referentes a cada laudo. Se o usuário clicar no ícone de download, será possível baixar um arquivo zip, contendo todas as imagens cadastradas referente ao laudo, além do Documento Word com os respectivos dados e seguindo a formatação mostrado na Seção 5.1.1. Assim como também, excluir um laudo.

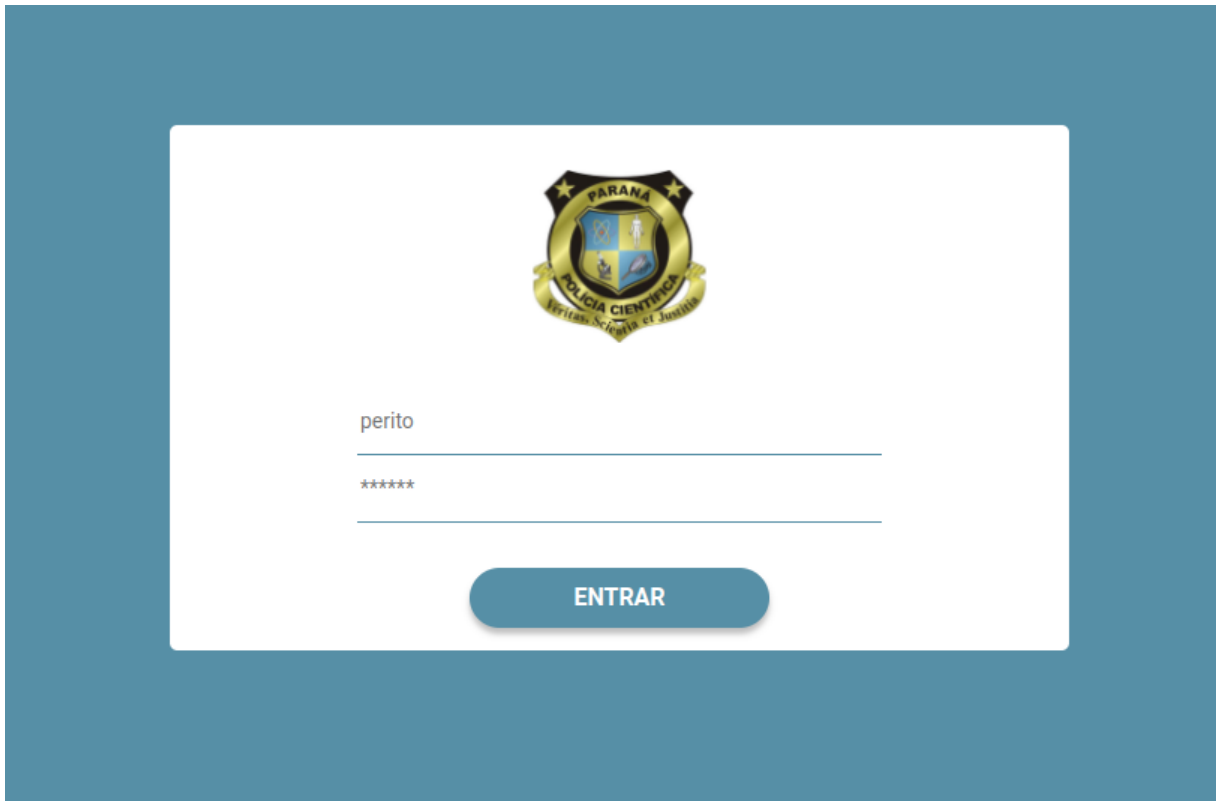


Figura 15 – Tela de Login
Fonte: Elaborado pela autora.

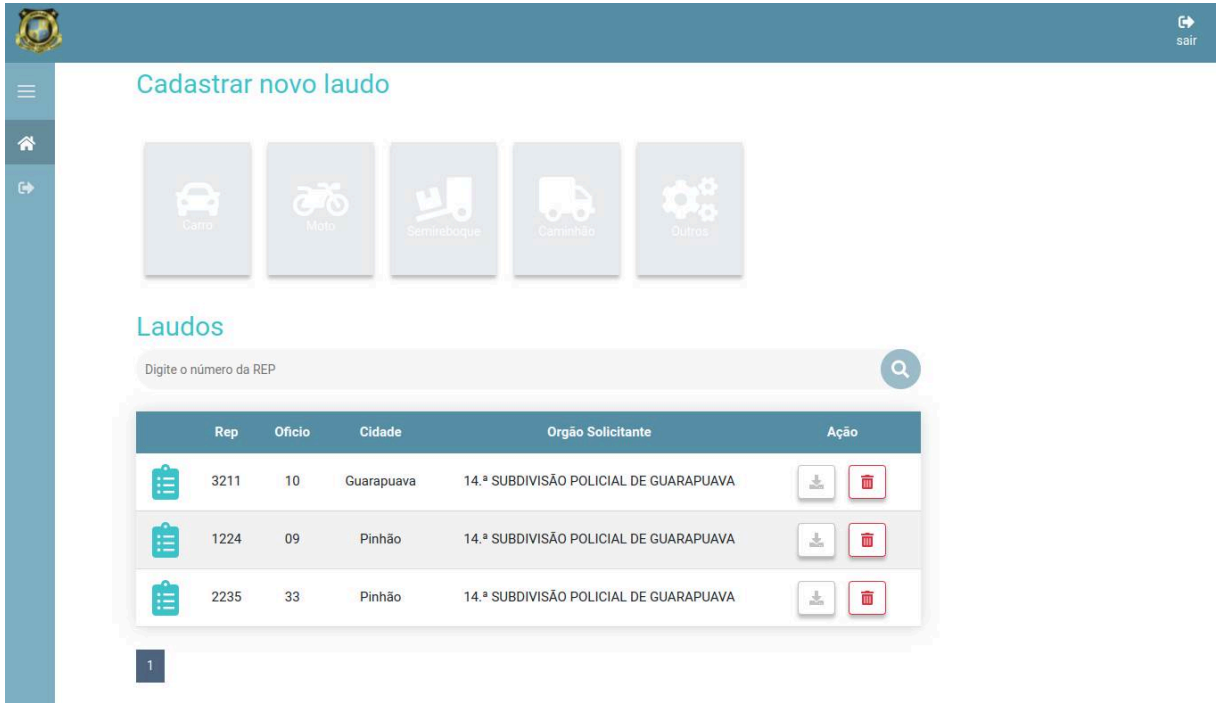


Figura 16 – Tela principal
Fonte: Elaborado pela autora.

5.3.3 CRIAÇÃO DO BANCO DE DADOS

Conforme exposto no Apêndice A, a especificação do banco de dados conta com uma grande diversidade de tabelas, atributos, e interligações. A fim de determinar a melhor maneira de construir essa estrutura e mantê-la organizada, foram realizados testes com duas bibliotecas, sendo que uma delas trabalha com SQL puro (mysql2), e a outra Sequelize⁴ (ORM - Object/Relational Mapper), uma biblioteca baseada em Promise para Node.js, que abstrai o gerenciamento dos dados. Utilizando a biblioteca mysql2, as queries mostraram-se mais complexas, já a biblioteca Sequelize facilitou o entendimento das mesmas, pois ela permite criar, buscar, alterar e remover dados do banco de dados utilizando métodos Javascript. Além disso, a biblioteca mostrou-se útil ao fornecer transações sólidas, facilidades em tratar relações e modificar as estruturas das tabelas, agilizando o processo de criação, validações, população, e migração de banco de dados. A Figura 17a apresenta o processo utilizando o Sequelize, e a Figura 17b ilustra a SQL pura utilizando mysql2.

```

3  module.exports = {
4    up: async (queryInterface, Sequelize) => {
5      return queryInterface.createTable('motorcycles', {
6        id: {
7          type: Sequelize.INTEGER,
8          primaryKey: true,
9          autoIncrement: true,
10         allowNull: false,
11        },
12        vehicle_id: {
13          type: Sequelize.INTEGER,
14          allowNull: false,
15          references: { model: 'vehicles', key: 'id' },
16          onUpdate: 'CASCADE',
17          onDelete: 'CASCADE',
18        },
19        report_id: {
20          type: Sequelize.INTEGER,
21          allowNull: false,
22          references: { model: 'reports', key: 'id' },
23          onUpdate: 'CASCADE',
24          onDelete: 'CASCADE',
25        },
26        created_at: {
27          type: Sequelize.STRING,
28          allowNull: false,
29        },
30        updated_at: {
31          type: Sequelize.STRING,
32          allowNull: false,
33        },
34      });
35    },
  }

```

```

db.execute(
`CREATE TABLE IF NOT EXISTS `reports`.`motorcycles` (
  `id` INT NOT NULL,
  `vehicle_id` INT NOT NULL,
  `report_id` INT NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_motorcycles_vehicles1_idx` (`vehicle_id` ASC),
  INDEX `fk_motorcycles_report1_idx` (`report_id` ASC),
  CONSTRAINT `fk_motorcycles_vehicles1`
    FOREIGN KEY (`vehicle_id`)
    REFERENCES `reports`.`vehicles` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_motorcycles_report1`
    FOREIGN KEY (`report_id`)
    REFERENCES `reports`.`report` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;`
)

```

Figura 17 – Método para Criar a Tabela Motorcycles Utilizando Sequelize.
Fonte: Elaborado pela autora.

⁴ Sequelize, website oficial: <https://bit.ly/3eRmbF1>

5.3.4 TESTES AUTOMATIZADOS

Afim de testar a integridade dos sistema e garantir o comportamento dos métodos implementados, foram desenvolvidos testes de integração e testes unitários. Para o desenvolvimento dos testes unitários foi utilizado o *framework Jest*, citado no referencial teórico na Seção 3.2.7.

Os testes tem como foco as principais funcionalidades do sistema, dentre elas: a verificação da criação do banco e a inserção dos laudos. Devido à dependência entre as tabelas, a cada laudo deve-se garantir que os todos os registros foram salvos em todas as tabelas adequadamente, caso contrário, se alguma informação for inválida, deve-se aplicar o processo de *rollback*, desfazendo todas as alterações (*inserts*) realizadas durante a transação. As figuras 18 e 19 apresentam exemplos de testes unitários para atributos válidos e inválidos, respectivamente.

A primeira figura ilustra um teste para inserir um laudo com exame íntegro. Um exame íntegro possui dois atributos: (1) *integrated*, que indica a integridade do exame; e (2), *number_chassi_original*, que representa a numeração identificadora do chassi. Se estas informações forem válidas, o laudo é inserido no banco e o teste verifica se as informações armazenadas no banco são iguais aos dados iniciais. Assim, a tabela de exames, que antes estava vazia, agora deve conter um registro.

Já na segunda figura, o teste espera que seja lançada uma exceção (throw exception) caso o exame íntegro não possua uma numeração identificadora (exigida no caso de exames íntegros). Assim, ao final do teste, a tabela de exames deve estar sem nenhum registro, com isso garante-se que foi realizado o *rollback* e os dados previamente inseridos foram removidos devido à falha na operação.

```

18   it('should get new instance in Exam Integrated', async () => {
19     let totalExame = await Exam.count();
20     expect(totalExame).toBe(0);
21
22     data = ExamFixture.getExamIntegrated();
23
24     const exam = await Exam.create(data);
25     expect(exam.integrated).toBe(data.integrated);
26     expect(exam.number_chassi_original).toEqual(data.number_chassi_original);
27
28     totalExame = await Exam.count();
29     expect(totalExame).toBe(1);
30   });

```

Figura 18 – Teste Unitário do Model Exame com Campos Válidos.

Fonte: Elaborado pela autora.

Foram desenvolvidos 47 testes no total, a Figura 20 demonstra a cobertura de testes da API. Este relatório é produzido pelo próprio Jest, que permite habilitar uma opção de *coverage*

nas configurações e ao executar os testes, o sistema entra em ação e gera um painel que permite visualizar trechos de código precisam de mais cobertura.

```

32  it('should launch a chassi throw exception is empty', async () => {
33    let totalExame = await Exam.count();
34    expect(totalExame).toBe(0);
35
36    data = ExamFixture.getExamDataChassiEmpty();
37
38    const expectError = () => {
39      return new ValidationError(
40        'Validation error: Parâmetros inválidos, insira uma numeração identificadora',
41      );
42    };
43    expect(Exam.create(data)).rejects.toThrow(expectError());
44    totalExame = await Exam.count();
45    expect(totalExame).toBe(0);
46  });
  
```

Figura 19 – Teste Unitário do Model Exame com Campos Inválidos.
 Fonte: Elaborado pela autora.

All files

63.77% Statements 2025/3175 90.62% Branches 145/160 91.39% Functions 85/93 63.77% Lines 2025/3175

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File	Statements	Branches	Functions	Lines
src	100%	59/59	100%	3/3
src/controllers	87.4%	465/532	80%	24/30
src/database	100%	66/66	100%	0/0
src/database/models	98.1%	932/950	93.24%	69/74
src/database/models/adulterated	88.88%	104/117	92.85%	13/14
src/database/models/factory	92.2%	71/77	66.66%	6/9
src/database/models/pieces	100%	161/161	100%	14/14
src/database/models/traits	97.05%	66/68	100%	8/8
src/doc	5.17%	57/1101	100%	0/0
src/errors	100%	8/8	100%	1/1
src/middlewares	100%	36/36	100%	7/7

Figura 20 – Cobertura de Testes.
 Fonte: Elaborado pela autora.

5.4 DESENVOLVIMENTO DO APLICATIVO MÓVEL

5.4.1 AMBIENTE DE DESENVOLVIMENTO

O desenvolvimento do aplicativo teve como princípio continuar trabalhando com JavaScript e ter um desenvolvimento híbrido, ou seja, desenvolver de uma forma que funcionasse tanto para Android quanto iOS. Dessa forma, a melhor maneira encontrada foi usar a plataforma Expo.io. A Figura 21 exemplifica o ambiente de desenvolvimento fornecido pela ferramenta, onde a aplicação executa no computador do desenvolvedor, e pode ser acessada remotamente no dispositivo móvel por meio da leitura de um QR Code.

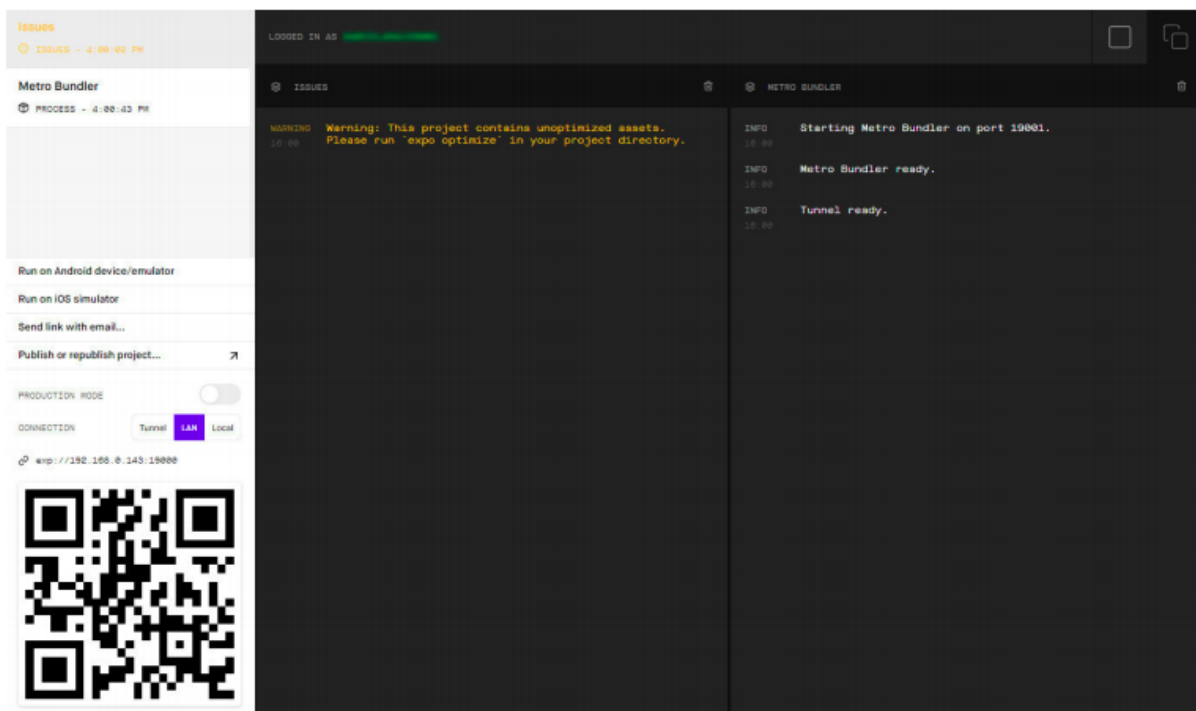


Figura 21 – Tela Inicial do Expo.

Fonte: Elaborado pela autora.

Juntamente com o Expo, foi utilizada a IDE de desenvolvimento Android Studio e o framework React Native com as bibliotecas react e Redux⁵, responsáveis pela organização e apresentação dos dados para o usuário.

5.4.2 CADASTRO DE LAUDOS

Conforme exposto no mapa conceitual da Seção 5.1.1, um laudo é composto por diferentes dados estruturados e dependentes. A segmentação da interface, característica do React, exige um mecanismo para compartilhamento de informações entre os diferentes componentes da aplicação. O Redux foi empregado como uma central de estados (*states*), gerenciando os

⁵ Redux, website oficial: <https://react-redux.js.org/>

dados dos usuários e dos laudos. Sem Redux as informações teriam que passar de componente em componente aumentando o acoplamento dos módulos da aplicação.

Assim, no aplicativo foi utilizado o Redux para auxiliar e facilitar esses tipos de ações, permitindo o compartilhamento de todas as informações salvas no *store* para todo o aplicativo. Com isso, a construção do formulário por etapas foi facilitada, permitindo que cada tela pudesse acessar informações das etapas anteriores, além as informações do login do usuário para compor o laudo.

O aplicativo possui três telas principais: Tela de Login a qual recebe o nome do usuário e a senha do perito. Ao executar o aplicativo pela primeira vez ou ainda, caso o usuário não esteja autenticado em sua conta, é exibida a tela de apresentação e em sequência, a tela de autenticação.

Após realizar a autenticação, os dados da conta são persistidos no dispositivo, não sendo mais necessário que o usuário se autentique em uma próxima execução do aplicativo a menos que ele limpe o cache da aplicação no dispositivo. Logo, após a tela de apresentação o usuário é encaminhado para tela inicial.

A Figura 22 apresenta a tela inicial em que o usuário pode o tipo de veículos periciado. Ao escolher um tipo de veículo, o usuário é encaminhado para tela de formulário, em que está dividido em cinco etapas: informações gerais, dados básicos, peças, exames e por fim, a última etapa, confirmação dos dados.

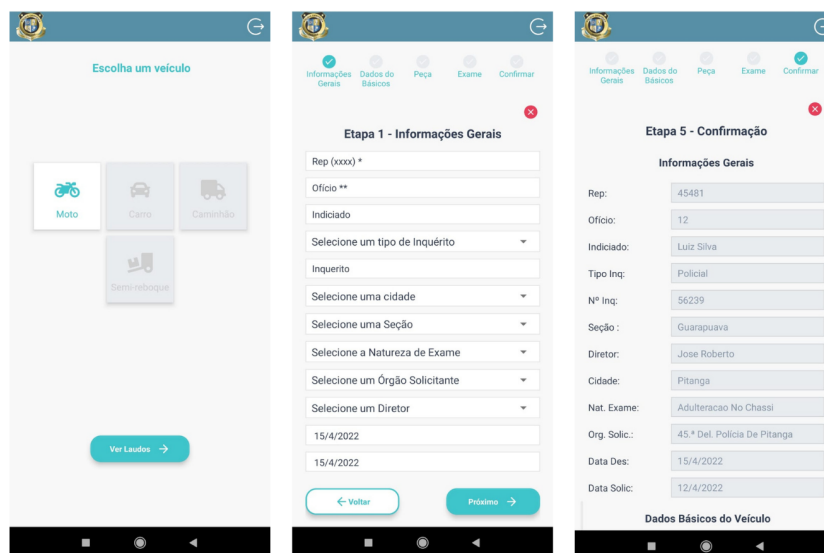


Figura 22 – Tela Inicial e Telas das Etapas 1 e 5 do Formulário.
Fonte: Elaborado pela autora.

O formulário do aplicativo, além das informações do Laudo Pericial foi implementado para permitir o registro de fotos do veículo durante o exame. Também foram previstas validações em alguns dos campos. Por exemplo, para a numeração da *rep* não é possível registrar com

menos de dois caracteres. Além dos preenchimentos dos campos, o usuário deve preencher todos os campos para prosseguir.

5.4.3 ENVIO DO LAUDO

Ao final do preenchimento dos dados, ao clicar no botão confirmar, as imagens salvas são codificadas em base64 e então o estado armazenado no Redux durante o preenchimento do formulário é transformado em JSON e as informações são armazenadas no recurso chamado AsyncStorage, uma das formas de persistência de dados do Expo. Posteriormente ao registro, o Laudo é disponibilizado na tela de laudos, onde é possível realizar uma chamada à API para o envio ao servidor e armazenado no banco de dados, ou a ainda a exclusão de um laudo.

5.4.4 EDIÇÃO E EXCLUSÃO DO LAUDO

O aplicativo também permite salvar e editar de um laudo incompleto. Ao tentar criar um novo laudo, caso exista um laudo incompleto, é apresentada uma lista de opções como demonstrado na Figura 23 (tela da esquerda), em que o usuário poderá optar por dar continuidade ou criar um novo. Ao escolher dar continuidade, o estado inicial no Redux recebe os dados armazenados no AsyncStorage do respectivo laudo, retornando as informações previamente preenchidas para aquele laudo. Para salvar um laudo incompleto, basta clicar no "x" em qualquer uma das etapas, são então exibidas as opções de "salvar e sair", "deletar e sair ou "cancelar" a ação, conforme apresentado na tela da direita da figura.

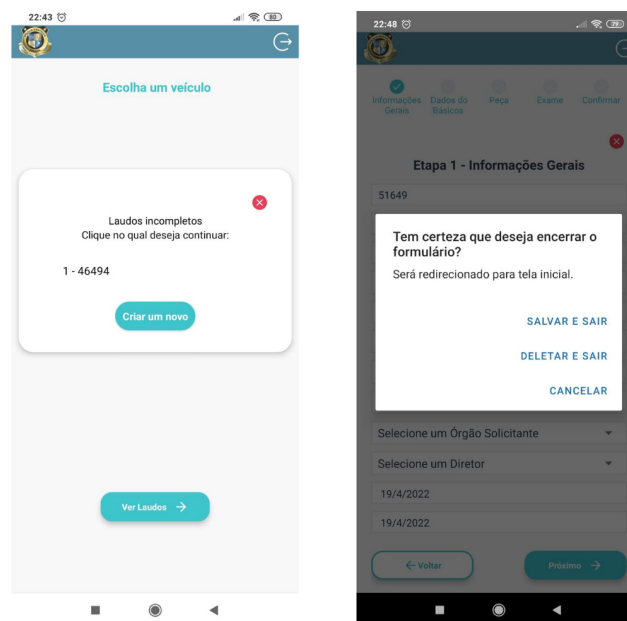


Figura 23 – Tela Inicial com Opção de Editar e Tela para Salvar um Laudo Incompleto.
Fonte: Elaborado pela autora.

5.4.5 DEPLOY

Após o desenvolvimento da API e da aplicação web, para fins de testes ambas foram alocadas em uma VM (Máquina Virtual, do inglês Virtual Machine) hospedada em um servidor gratuito de plano *Always Free* na *Oracle Cloud*.

Ao realizar o *deploy*, foi necessária a instalação e a criação do banco dados. Para auxiliar neste processo, foi utilizado o Docker, uma plataforma aberta que facilita a criação e administração de contêineres ou ambientes isolados. Para a definição do contêiner foi utilizada a ferramenta de Docker-Compose, que possibilita a definição e configuração de um ou mais contêineres dentro da mesma aplicação em um mesmo arquivo, além dos *scripts* para fazer o *build* e execução de forma mais direta. Para o banco de dados foi utilizada a imagem Docker oficial do MYSQL.

Uma das vantagens em utilizar o Docker, foi que com essa ligação e facilidade, as configurações ficam portáteis para qualquer outro ambiente que também possua o Docker instalado. A consequência é a redução do tempo de *deploy* da aplicação. Assim, é desnecessário ajustar o ambiente para que o serviço funcione corretamente, ou seja sempre será igual, sendo configurado uma única vez.

Para inicialização e monitoramento do servidor em produção foi utilizado o PM2 (Advanced, Production Process Manager) que é um gerenciador de processos para Node.js que possibilita administrar o funcionamento das aplicações, além de organizar as aplicações que estão rodando no servidor também foi utilizado para gerenciamento de *logs* e erros.

Por fim, para exposição da aplicação na Internet, foi necessário utilizar o servidor web NGINX para administrar as requisições recebidas. NGINX é responsável por fazer o proxy reverso para cada uma das aplicações, nomeadas "api" e "web". Assim o único serviço utilizando a porta 80 passa a ser o NGINX, e as requisições são tratadas como subdomínios: "web" redireciona para a aplicação estática desenvolvida em React JS; "api" redireciona chamadas para o servidor Node.js.

6 CONSIDERAÇÕES FINAIS

Este trabalho foi realizado com o objetivo de prover um meio tecnológico para substituição e melhoria do processo convencional existente hoje para a geração de Laudos Veiculares do Instituto de Criminalística. O objetivo da implementação foi prover um MVP para a solução deste problema, diante disso, foi desenvolvido um sistema de coleta de dados automatizado, que possui três funções primordiais: a confecção de laudos veiculares por meio de um formulário em etapas com a coleta dos dados pelo aplicativo móvel, a transferência destes dados para a aplicação Web via Intranet e, por fim, o gerenciamento de laudo no formato exigido de forma fácil e intuitiva.

Diante do contexto proposto para este trabalho, os requisitos funcionais foram implementados, resultando um sistema capaz de coletar e gerar laudos veiculares para motocicletas. A arquitetura do sistema foi especificada e exposta juntamente com a modelagem do banco de dados, protótipos de telas, e um mapa conceitual, que favorecem a expansão do sistema para laudos de outros tipos de veículos.

Destaca-se que a aplicação conta com testes automatizados das principais funcionalidades, e que o acesso ao aplicativo para a coleta de dados pode ser realizado de maneira offline, de acordo com as exigências do Instituto de Criminalística. Com isso, o perito consegue confeccionar o laudo mesmo sem acesso à Internet e, posteriormente, ao chegar na unidade efetuar o envio dos laudos do dispositivo móvel para o sistema Web e então realizar o download do laudo completo, em formato Docx, incluindo as imagens.

6.1 TRABALHOS FUTUROS

Esta seção lista algumas possibilidades de trabalhos futuros. As funcionalidades citadas podem tornar o sistema mais completo e seguro, permitindo inclusive que se torne um produto aplicável em produção.

- Facilitar o cadastro de mais usuários administradores ou peritos (sem a necessidade de acesso ao servidor back-end);
- Permitir o envio dos dados da aplicação móvel para API somente via Intranet;
- Permitir a edição dos dados na aplicação web, o sistema atualmente não permite a edição dos dados do laudo na aplicação web;
- Permitir uma pré-visualização do laudo antes de efetuar o download, o sistema deve permitir a pré-visualização do laudo completo;
- Salvar as imagens em um servidor dedicado, por exemplo AWS;
- Expansão para outros tipos de laudos e tipo de veículos;

- Implementar o espelhamento do banco de dados do servidor no aplicativo móvel;
- Investigar o uso da Context API ao invés de Redux no aplicativo.

REFERÊNCIAS

- AMORIM, M. *et al.* Assistente de cena de crime: Solução para auxílio à investigação de homicídios. In: SBC. **Anais do II Workshop Brasileiro de Cidades Inteligentes**. [S.l.], 2019.
- AZEVEDO, L. J. d. M. de; MIAZAKI, M.; PORFIRIO, A. J. Questionário eletrônico em ambiente android para coleta de dados. 2016.
- CANTO, J. R. A. do. **Plataforma para a Coleta de Dados para o acompanhamento de processos administrativo**. 2017. Disponível em: <https://comunidade.startse.com/in/e-laudo>. Acesso em: 15/11/2020.
- CRIMINALISTICA. **Identificação de Veículos**. 2019. Disponível em: <http://www.ic.pr.gov.br/modules/conteudo/conteudo.php?conteudo=29>. Acesso em: 02/03/2020.
- DETRAN. **Vistoria digital de veículos é mais rápida e segura**. 2016. Acesso em: 24/04/2020.
- EISENMAN, B. **Learning react native: Building native mobile apps with JavaScript**. [S.l.]: "O'Reilly Media, Inc.", 2015.
- EXPO. **Expo: a framework and a platform for universal React applications**. 2020. Disponível em: <https://docs.expo.io/>. Acesso em: 08/10/2020.
- EXPRESS. **Express**. 2017. Disponível em: <http://expressjs.com>. Acesso em: 18/07/2021.
- FACEBOOK. **Jest: Delightful JavaScript Testing**. 2017. Disponível em: <http://facebook.github.io/jest/>. Acesso em: 02/03/2021.
- FREEMAN, E. *et al.* **Use A Cabeça Padrões E Projetos**. [S.l.]: Alta Books, 2009.
- HAHN, E. **Express in Action: Writing, building, and testing Node. js applications**. [S.l.]: Manning Publications, 2016.
- MOGNON, M. Sistema para gerenciar a elaboração de laudos periciais balísticos. 2019. Disponível em: <https://repositorio.utfpr.edu.br/jspui/handle/1/23602>.
- PRADO. **O laudo pericial diante dos processos judiciais**. 2014. Disponível em: <https://jus.com.br/artigos/31779/o-laudo-pericial-diante-dos-processos-judiciais>. Acesso em: 09/03/2020.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016.
- REACT. **NA JavaScript library for building user interfaces**. 2020. Disponível em: <https://reactjs.org/>. Acesso em: 04/10/2020.
- SMARTBEAR, S. **About Swagger Specification Documentation**. 2020. Disponível em: <https://swagger.io/specification>. Acesso em: 10/04/2022.
- SOMMERVILLE, I. **Engenharia de software**. ADDISON WESLEY BRA, 2008. ISBN 9788588639287. Disponível em: <https://books.google.com.br/books?id=iflYOgAACAAJ>.
- SSP. **Relatório Estatístico Criminal 2019**. 2019. Disponível em: <https://www.seguranca.pr.gov.br/CAPE/Estatisticas>. Acesso em: 21/04/2020.
- WIKIPEDIA. **ECMAScript**. 2021. Disponível em: <https://en.wikipedia.org/wiki/ECMAScript#ES2021>. Acesso em: 10/04/2022.

APÊNDICE A – Diagrama Entidade-Relacionamento

O diagrama Entidade-Relacionamento é representado nas figuras 24 e 25. Ambas as figuras representam uma mesma modelagem ¹ onde as conexões entre as entidades foram representadas pelas letras A, B, C, e D.

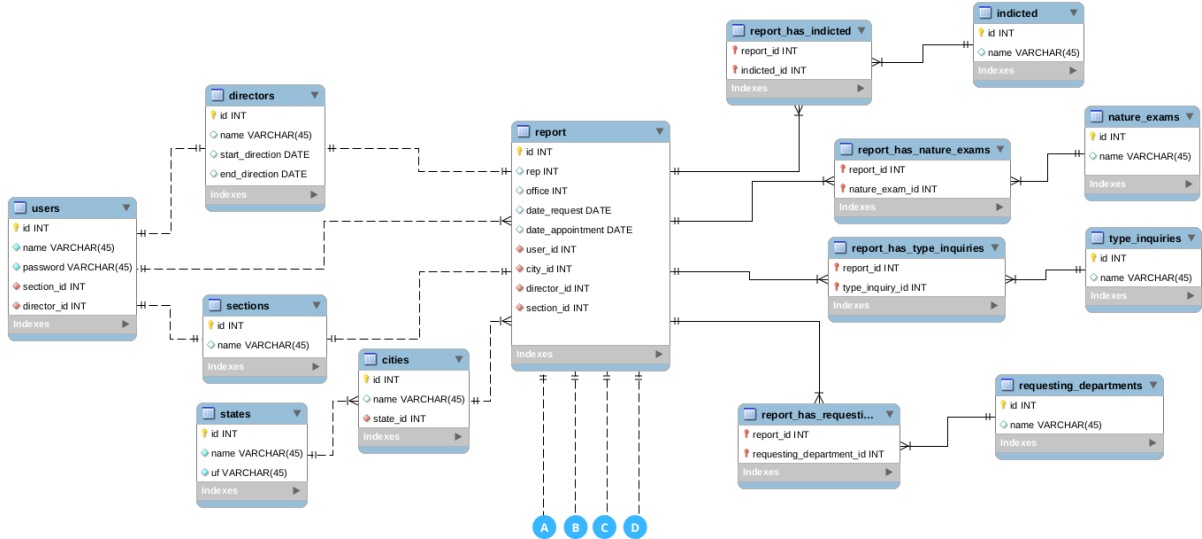


Figura 24 – Diagrama Entidade Relacionamento do Banco, parte 1. Fonte: Elaborado pela autora.

¹ A imagem completa do modelo físico do banco de dados pode ser encontrada em <http://bit.ly/DiagramaBanco-tcc-amanda>

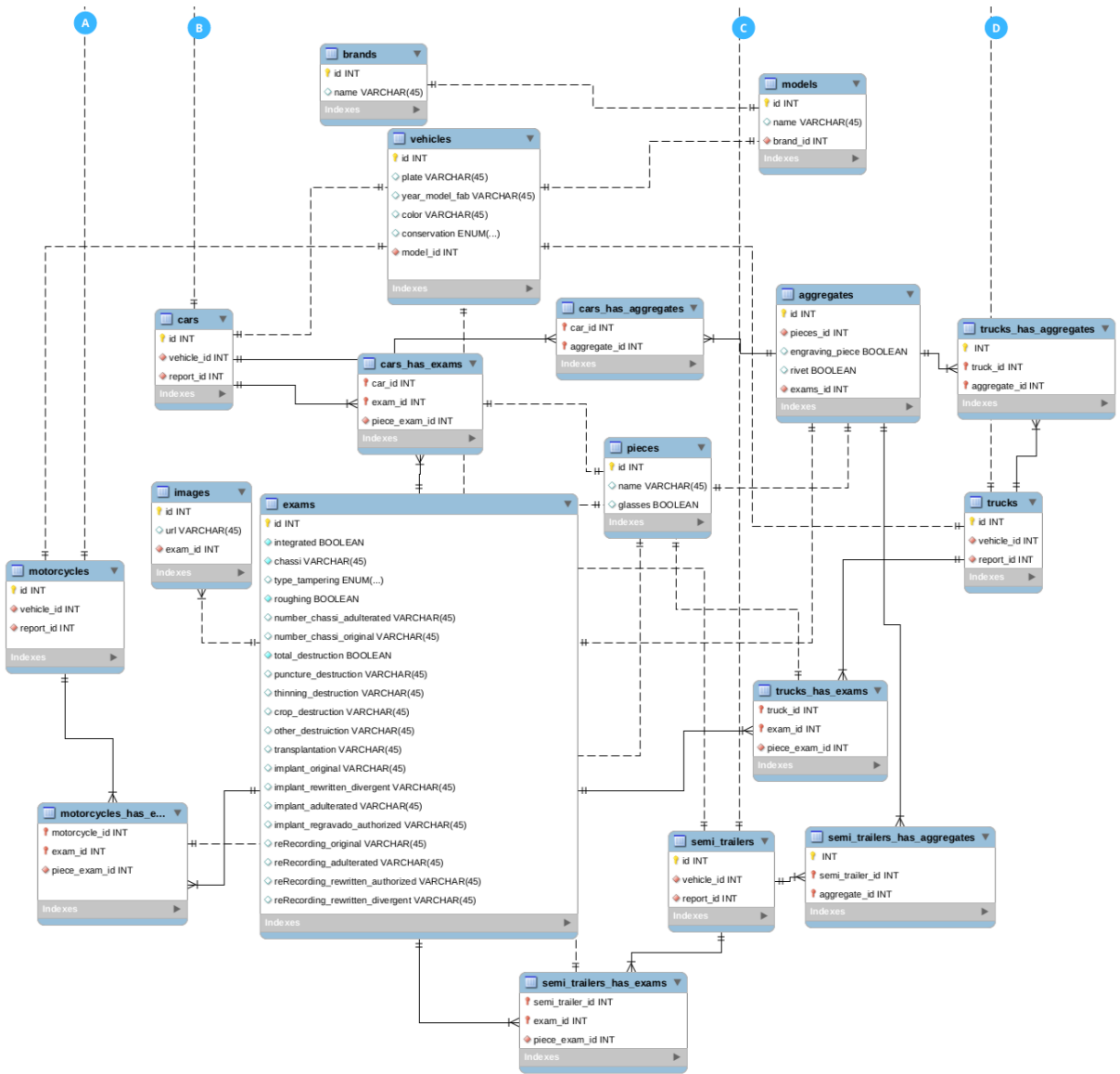


Figura 25 – Diagrama Entidade-Relacionamento do Banco, parte 2. Fonte: Elaborado pela autora.

APÊNDICE B – Documento Word - Laudo

Documento Word com peças íntegras gerado pela aplicação web é representado nas Figuras 26 e 27. Ambas as figuras representam um mesmo laudo.

FLS. 2
LAUDO N° 18868

LAUDO DE EXAME DE VEÍCULO A MOTOR

Aos **6 de março de 2022**, nesta cidade de **Guarapuava** e no **INSTITUTO DE CRIMINALÍSTICA** do Estado, foi designado pelo Diretor do Instituto, **José Roberto**, por indicação do Chefe da Seção, o Perito Criminal **José Roberto**, para proceder ao exame no veículo mencionado no **Ofício sob o N° 140**, recebido dia **7 de março de 2022**, oriundo de **14.ª SUBDIVISÃO POLICIAL DE GUARAPUAVA**.

Em consequência, o Perito procedeu ao exame solicitado, relatando-o com verdade e com todas as circunstâncias, da forma como segue:

MOTIVO DA PERÍCIA

Depreende-se da leitura do ofício supracitado, que a perícia tem por finalidade proceder ao exame nas numerações identificadoras do veículo acima mencionado.

DO EXAME

O veículo apresentado a exame foi periciado no pátio da delegacia solicitante. Trata-se de uma **Moto** de marca de fabricação **Honda**, de cor **azul**, desprovida de placa de licenciamento e em **Bom** estado de conservação ostentando placas de licenciamento **abc-2317, Pinhão**.



VEÍCULO PERICIADO

Figura 26 – Documento Word Gerado pela aplicação, parte 1 - Cabeçalho do laudo. Dados reais e informações sensíveis foram omitidos nesta imagem.

Fonte: Elaborado pela autora.

Documento Word com peça adulterada e outra íntegra é demonstrado nas Figuras 28 e 29. Ambas as figuras representam um mesmo laudo.

O referido veículo apresenta a seguinte numeração identificadora: **2AEHD5RPA0KHK3894**. E número do motor **2AEHD5RPA0KHK3432**. Tendo-se procedido ao acurado exame nas referidas numerações, bem como nas superfícies que lhe servem de suporte, **não foram observados quaisquer sinais ou vestígios de adulteração.**



NUMERAÇÃO DO CHASSI



NUMERAÇÃO DO MOTOR

Este laudo foi redigido pelo Perito **José Roberto** e disponibilizado em arquivo digital contendo uma folha de rosto dois e disponibilizado em arquivo digital contendo uma folha de rosto e mais uma página. Por nada mais haver e sendo essas as declarações que tem a fazer, deu-se por findo o exame solicitado que de tudo se lavrou o presente laudo, o qual segue digitalmente assinado.

Figura 27 – Documento Word Gerado pela aplicação, parte 2 - Exames das peças chassi e motor de uma moto. Dados reais e informações sensíveis foram omitidos nesta imagem.

Fonte: Elaborado pela autora.

FLS. 2
LAUDO Nº 4445

LAUDO DE EXAME DE VEÍCULO A MOTOR

Aos **1 de junho de 2022**, nesta cidade de **Guarapuava** e no **INSTITUTO DE CRIMINALÍSTICA** do Estado, foi designado pelo Diretor do Instituto, **Carlos Silva**, por indicação do Chefe da Seção, o Perito Criminal **José Roberto**, para proceder ao exame no veículo mencionado no **Ofício sob o Nº 888**, recebido dia **8 de junho de 2022**, oriundo de **45.ª DELEGACIA REGIONAL DE POLÍCIA DE PITANGA**.

Em consequência, o Perito procedeu ao exame solicitado, relatando-o com verdade e com todas as circunstâncias, da forma como segue:

MOTIVO DA PERÍCIA

Depreende-se da leitura do ofício supracitado, que a perícia tem por finalidade proceder ao exame nas numerações identificadoras do veículo acima mencionado.

DO EXAME

O veículo apresentado a exame foi periciado no pátio da delegacia solicitante. Trata-se de uma **Moto** de marca de fabricação **Honda**, de cor **vermelha**, desprovida de placa de licenciamento e em **Bom** estado de conservação ostentando placas de licenciamento **ABC-1234, Pitanga**.



VEÍCULO PERICIADO

Para melhor ordenamento dos trabalhos, a parte relativa ao exame obedecerá à seguinte orientação:

Figura 28 – Documento Word Gerado pela aplicação, parte 1 - Cabeçalho do laudo. Dados reais e informações sensíveis foram omitidos nesta imagem.

Fonte: Elaborado pela autora.

I. NUMERAÇÃO DO CHASSI

Essa numeração, na **Moto** periciada, acha-se inscrita no canote do guidão no lado direito. Ao exame do suporte onde se acha gravada a numeração do chassi, verificou o Perito, evidentes sinais deixados pela operação ali procedida que consistiu no Desbaste acarretando a destruição da numeração original. Submetida a superfície em referência ao tratamento químico-metalográfico, destinado a revelar remanescentes da gravação primitiva, observou-se a sequência **KE2 RV815GfTf57323**



NUMERAÇÃO DO CHASSI

II. NUMERAÇÃO DO MOTOR

Na continuidade do exame, o perito verificou que a numeração do Motor **KE2 RV815GfTf57865**. Tendo-se procedido ao acurado exame na referida numeração, bem como na superfície que lhe serve de suporte, **não foram observados quaisquer sinais ou vestígios de adulteração.**



NUMERAÇÃO DO MOTOR

Este laudo foi redigido pelo Perito **José Roberto** e disponibilizado em arquivo digital contendo uma folha de rosto dois e disponibilizado em arquivo digital contendo uma folha de rosto e mais duas páginas. Por nada mais haver e sendo essas as declarações que tem a fazer, deu-se por findo o exame solicitado que de tudo se lavrou o presente laudo, o qual segue digitalmente assinado.

Figura 29 – Documento Word Gerado pela aplicação, parte 2 - Exames das peças chassi e motor de uma moto. Dados reais e informações sensíveis foram omitidos nesta imagem.

Fonte: Elaborado pela autora.