VICTOR MOREIRA GOMES

SIGMABEL – SISTEMA DE GERENCIAMENTO DE MATERIAIS BÉLICOS

Ji-Paraná
2023

VICTOR MOREIRA GOMES

SIGMABEL – SISTEMA DE GERENCIAMENTO DE MATERIAIS BÉLICOS

Monografia apresentada à Banca Examinadora do curso de Sistemas de Informação do Centro Universitário São Lucas Ji-Paraná, como requisito de aprovação para obtenção do Título de Bacharel em Sistemas de Informação.

Orientador: Prof. Esp. Wallison Storck Coelho

Ji-Paraná
2023

Dados Internacionais de Catalogação na Publicação – CIP

G 633 Gomes, Victor Moreira

SIGMABEL – Sistema de Gerenciamento de Materiais Bélicos / Victor Moreira Gomes – Rondônia. 2023.
78.f; 30cm.

Trabalho de Conclusão de Curso (Graduação) Centro Universitário São Lucas Ji-Paraná, Curso de Sistemas de Informação, 2023.

Orientador: Prof. Esp. Wallison Storck Coelho

1. Gerenciamento de materiais. 2. Informatização. 3. Gestão de materiais 4. Tecnologia. I. Título II. Coelho, Wallison Storck. III. Título.

CDD 355.07

Ficha Catalográfica Elaborada pela Bibliotecária Dávilla V.O. da Silva CRB11/954

RESUMO

Este trabalho propõe o desenvolvimento do SIGMABEL (Sistema de Gerenciamento de Materiais Bélicos), um software web que visa substituir o processo manual realizado em livro ata, proporcionando maior segurança e eficiência na gestão de materiais bélicos. O SIGMABEL um software web desenvolvido utilizando as linguagens PHP e HTML, com o auxílio do framework Laravel e a aplicação dos padrões de MVC. O objetivo do software é oferecer uma interface simples e de fácil utilização, proporcionando ao gestor as informações necessárias para um planejamento estratégico e eficaz. Com isso, espera-se auxiliar na tomada de decisões mais assertivas e minimizar os riscos envolvidos no gerenciamento de armas, coletes, munições e outros materiais utilizados por uma Organização Policial Militar no cumprimento da preservação da ordem pública e da segurança das pessoas e do patrimônio. Ao automatizar esse processo por meio do SIGMABEL, espera-se aumentar o poder de gerenciamento e garantir maior controle sobre o inventário dos materiais bélicos, contribuindo para uma gestão mais eficiente e segura.

Palavras-chave: Tecnologia, automatização, controle de material bélico, gerenciamento, Polícia Militar.

ABSTRACT

This project proposes the development of SIGMABEL (Weaponry Material Management System), a web-based software aimed at replacing the manual process carried out in a logbook, thus providing greater security and efficiency in the management of weaponry materials. SIGMABEL is a web-based software developed using PHP and HTML languages, with the assistance of the Laravel framework and the application of MVC design patterns. The software's objective is to offer a simple and user-friendly interface, providing managers with the necessary information for strategic and effective planning. It is expected to assist in making more accurate decisions and minimize the risks involved in managing firearms, body armors, ammunition, and other materials used by a Military Police Organization in the preservation of public order and the security of individuals and assets. By automating this process through SIGMABEL, it is expected to enhance management capabilities and ensure greater control over the inventory of weaponry materials, thereby contributing to a more efficient and secure management system.

Keywords: Technology, automation, ordnance control, management, Military Police.

LISTA DE ILUSTRAÇÕES

Figura 1 - Distribuição dos Batalhões em Rondônia 10
Figura 2 - Pistola PT-100 utilizada pela PMRO 13
Figura 3 - Exemplo de materiais bélicos utilizados pelas Polícias Militares 14
Figura 4 - Informatização de um processo manual 15
Figura 5 - Diagrama de desenvolvimento de um sistema de informação 16
Figura 6 - Comando de criação de um projeto Laravel 23
Figura 7 - Processo do MVC 24
Figura 8 - Visão geral do programa Laragon 25
Figura 9 - Visão geral do programa Visual Studio Code 27
Figura 10 - Componentes de interface do framework Bootstrap 28
Figura 11 - Visão geral do programa HeidiSQL 30
Figura 12 - Mapa mental das vantagens do MySQL 31
Figura 13 - Desenvolvimento de uma solução de sistema de informação 34
Figura 14 - Diagrama de caso de uso 39
Figura 15 - Diagrama de classes 40
Figura 16 - Diagrama de atividade UC01 41
Figura 17 - Diagrama de atividade UC02 42
Figura 18 - Diagrama de atividade UC03 42
Figura 19 - Diagrama de atividade UC04 43
Figura 20 - Diagrama de atividade UC05 44
Figura 21 - Diagrama de atividade UC06 44
Figura 22 - Diagrama de atividade UC07 45
Figura 23 - Diagrama de atividade UC08 45
Figura 24 - Diagrama de entidade relacionamento 47

Figura 25 - Estrutura de pastas do projeto 48
Figura 26 - Tela inicial da aplicação 49
Figura 27 - Tela de cadastro do usuário 53
Figura 28 - Tela de login 54
Figura 29 - Tela de informações e avisos 54
Figura 30 - Tela dos tipos de materiais cadastrados 55
Figura 31 - Tela para cadastro do tipo de material 55
Figura 32 - Tela dos materiais cadastrados 56
Figura 33 - Tela para cadastro de material 56
Figura 34 - Tela para edição de material 57
Figura 35 - Tela dos policiais cadastrados 57
Figura 36 - Tela das cautelas cadastradas 58
Figura 37 - Tela para cadastro de cautela 58
Figura 38 - Tela de avisos 59
Figura 39 - Tela cadastro de aviso 59
Figura 40 - Tela de emissão de relatórios 60
Figura 41 – Relatório por armeiro emitido em PDF 61
Figura 42 - Relatório por material emitido em PDF 61

LISTA DE QUADROS

Quadro 1 - Requisitos Funcionais 36
Quadro 2 - Requisitos Não-Funcionais 37
Quadro 3 - Dicionário de dados da entidade avisos 67
Quadro 4 - Dicionário de dados da entidade cautelas 67
Quadro 5 - Dicionário de dados da entidade failed_jobs 68
Quadro 6 - Dicionário de dados da entidade materials 68
Quadro 7 - Dicionário de dados da entidade migrations 68
Quadro 8 - Dicionário de dados da entidade model_has_permissions 69
Quadro 9 - Dicionário de dados da entidade model_has_holes 69
Quadro 10 - Dicionário de dados da entidade password_reset_tokens 69
Quadro 11 - Dicionário de dados da entidade permissions 69
Quadro 12 - Dicionário de dados da entidade personal_access_tokens 70
Quadro 13 - Dicionário de dados da entidade policials 70
Quadro 14 - Dicionário de dados da entidade role_has_permissions 70
Quadro 15 - Dicionário de dados da entidade roles 71
Quadro 16 - Dicionário de dados da entidade tipo_materials 71
Quadro 17 - Dicionário de dados da entidade users 71

LISTA DE ABREVIATURAS E SIGLAS

BPM – Batalhão de Polícia Militar

CAC – Caçadores, Atiradores e Colecionadores

CRUD – *Create, Read, Update, Delete*

CSS – *Cascading Style Sheets*

CS – *Comma Separated Values*

DDL – *Data Definition Language*

D-LOG – Departamento Logístico do Exército Brasileiro

DML – *Data Manipulation Language*

ER – *Entity Relationship*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IDE – *Integrated Development Environment*

MVC – *Model, View e Controller*

OPM – Organização Policial Militar

PDF – *Portable Document Format*

PHP – *Personal Home Page (Acrônimo de Hypertext Preprocessor)*

PM – Polícia Militar

PMRO – Polícia Militar do Estado de Rondônia

PNG – *Portable Network Graphics*

RE – Registro Estatístico

SGBD – Sistema Gerenciador de Banco de Dados

SGBDR – Sistema Gerenciador de Banco de Dados Relacional

SIG – Sistemas de Informação Gerenciais

SIGMA – Sistema de Gerenciamento Militar de Armas

SINARM – Sistema Nacional de Armas

SQL – *Structured Query Language*

SSH – *Secure Socket Shell*

SSL – *Secure Sockets Layer*

TLS – *Transport Layer Security*

UML – *Unified Modeling Language*

URL – *Uniform Resource Locator*

WAMP – Windows, Apache, MySQL e PHP

XML – *eXtensible Markup Language*

XP – *eXtreme Programming*

SUMÁRIO

1 INTRODUÇÃO 6

1.1 Problematização 6
1.2 Hipóteses 7
1.3 Objetivos 7
1.3.1 Objetivo Geral 7
1.3.2 Objetivos Específicos 7
1.4 Justificativa 8

2 REFERENCIAL TEÓRICO 9

2.1 Polícia Militar do Estado de Rondônia 9
2.2 Aspectos Legais 10
2.3 Materiais Bélicos 12
2.4 Informatização do Processo 14
2.4.1 Sistemas de Informação Gerenciais 15
2.4.2 PHP 17
2.4.3 HTML 18
2.4.4 UML 19
2.4.5 SQL 20

3 MATERIAIS E MÉTODOS 22

3.1 Materiais 22
3.1.1 Laravel 10 22
3.1.1.1 MVC - Model-View-Controller 23
3.1.2 Laragon 25
3.1.2.1 Apache 26
3.1.3 Visual Studio Code 26
3.1.4 Bootstrap 28
3.1.5 HeidSQL 29
3.1.5.1 MySQL 30
3.1.6 Google Chrome 31
3.1.7 Draw.io 32
3.2 Métodos 33
3.2.1 Metodologia eXtreme Programming - XP 34
3.2.2 Requisitos do Sistema 35

3.2.2.1 Requisitos Funcionais 36
3.2.2.2 Requisitos Não Funcionais 37
3.2.3 Modelagem do Sistema 38
3.2.3.1 Diagrama de Caso de Uso 39
3.2.3.2 Diagrama de Classe 40
3.2.3.3 Diagramas de Atividade 40
3.2.3.4 Diagrama de Entidade Relacionamento 46
4 RESULTADOS 48
4.1 Metodologia e Padrões Escolhidos 50
4.2 Sistema Desenvolvido 51
4.3 Relatórios Gerados Pelo Sistema 60
5 CONCLUSÃO 62
REFERÊNCIAS 64
APÊNDICE A – DICIONÁRIO DE DADOS 67

1 INTRODUÇÃO

Atualmente a tecnologia está cada vez mais inserida em nossas atividades diárias: nos celulares, veículos, televisões e até nos eletrodomésticos. E quando observamos o comércio, a indústria e a administração pública, é possível ver a informática com um papel fundamental para o desenvolvimento e expansão das melhorias desses setores. Sommerville (2011, p. 6) afirma que “negócio e sociedade estão mudando de maneira incrivelmente rápida, à medida que as economias emergentes se desenvolvem e as novas tecnologias se tornam disponíveis”. A automatização de processos e o controle de atividades estão totalmente ligados à melhoria, produtividade e a redução de custos. Gouveia (2023) comenta que os processos manuais quando são automatizados tendem a progredir e estão totalmente ligados com a avanço da gestão como um todo.

Devido a minha profissão Policial Militar, visualizando as operações diárias enfrentadas para o controle do cautelamento dos materiais bélicos faço deste projeto cujo objetivo é alcançar melhores resultados, incluindo a tecnologia dos Sistemas de Informação no procedimento de controle da reserva de armamento de materiais bélicos da Polícia Militar, que exige um controle rigoroso e preciso das movimentações internas de cada material. Esta automatização proporcionará um melhor funcionamento operacional relacionado à movimentação, de maneira eletrônica do acervo garantindo maior controle e agilidade desde o cautelamento para serviço até a devolução.

1.1 Problematização

A Polícia Militar do Estado de Rondônia possui um vasto material bélico empregado no policiamento ordinário, que inclui armas, coletes, munições e demais materiais. E todo esse material necessita de um controle e armazenamento especial, registro histórico de movimentação e registro de manutenções, nos casos em que houver. No entanto a gestão de todo esse patrimônio ainda é feita manuscrita em papéis: livro ata, não sendo possível emitir relatórios ou ter conhecimento de imediato a informações sem antes realizar uma contagem nos livros. Isso dificulta o as tomadas de decisões, podem ocorrer erros, falhas, perda de dados que podem ocasionar retrabalho. É perceptível que a gestão empregada não é compatível com as

exigências requeridas, onde os dados estão suscetíveis às intempéries, rasuras e extravios.

1.2 Hipóteses

Com recursos reduzidos e muitos órgãos públicos para gerir, a falta de desenvolvimento de ferramentas de gestão de patrimônio informatizado por parte do governo do estado faz com que muitos processos aconteçam de forma manual em papéis arquivados em salas de arquivo, em casos isolados são controlados por meio de planilhas eletrônicas de forma descentralizada e sem gestão. O uso da tecnologia com ferramentas de desenvolvimento traz esse aporte ao órgão do estado, em específico a Polícia Militar, onde visa auxiliar a gestão com o uso *software* desenvolvido para gestão e controle do acervo do material bélico.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um sistema que gerencie o inventário e controle as cautelas de armas, coletes, munições e demais materiais bélicos de carga utilizados no policiamento ordinário, baseado metodologia já utilizada na Polícia Militar do Estado de Rondônia.

1.3.2 Objetivos Específicos

- Desenvolver um *software web* em PHP (Personal Home Page) através do ambiente de desenvolvimento Laragon utilizando o *framework* Laravel para gerenciar o patrimônio bélico da OPM (Organização Policial Militar);
- Desenvolver um módulo de cadastro e controle dos materiais bélicos (armas, munições, coletes, rádios, etc) utilizados em uma unidade policial;
- Desenvolver um módulo de cadastro e controle dos policiais da unidade que utilizam os materiais para o serviço de policiamento ordinário;
- Desenvolver um módulo de controle dos empréstimos diários (cautelas) para uso no serviço policial ordinário;

- Desenvolver uma *dashboard* que possibilite a visualização de dados dos materiais e dos empréstimos de forma rápida e simples em tempo real;

- Disponibilizar a emissão de relatórios para auxiliar as possíveis tomadas de decisões.

1.4 Justificativa

A morosidade de executar um processo manual de cautelamento do material bélico para saída ao serviço operacional pode acontecer de maneira mais rápida e eficiente através de um processo automatizado, assegurando a exatidão e segurança que a informatização pode entregar. Além disso, o armazenamento digital das informações de empréstimo (cautela) irá facilitar as eventuais consultas que poderão acontecer onde serão gerados relatórios do acervo disponível e dos empréstimos realizados.

A escolha de uma aplicação *web* para o desenvolvimento desse projeto se justifica por ser um *software* de fácil execução em computadores simples que não exigem muito da máquina, e ter a possibilidade de ser acessado na rede por todo escalão hierárquico onde será feita a aplicação do sistema. Por se tratar da gestão de armas de fogo os gestores militares, poderão dar fiel cumprimento do controle e registro do material bélico sob sua responsabilidade, eliminando a necessidade do registro manual de dados, reduzindo a possibilidade de erros humanos.

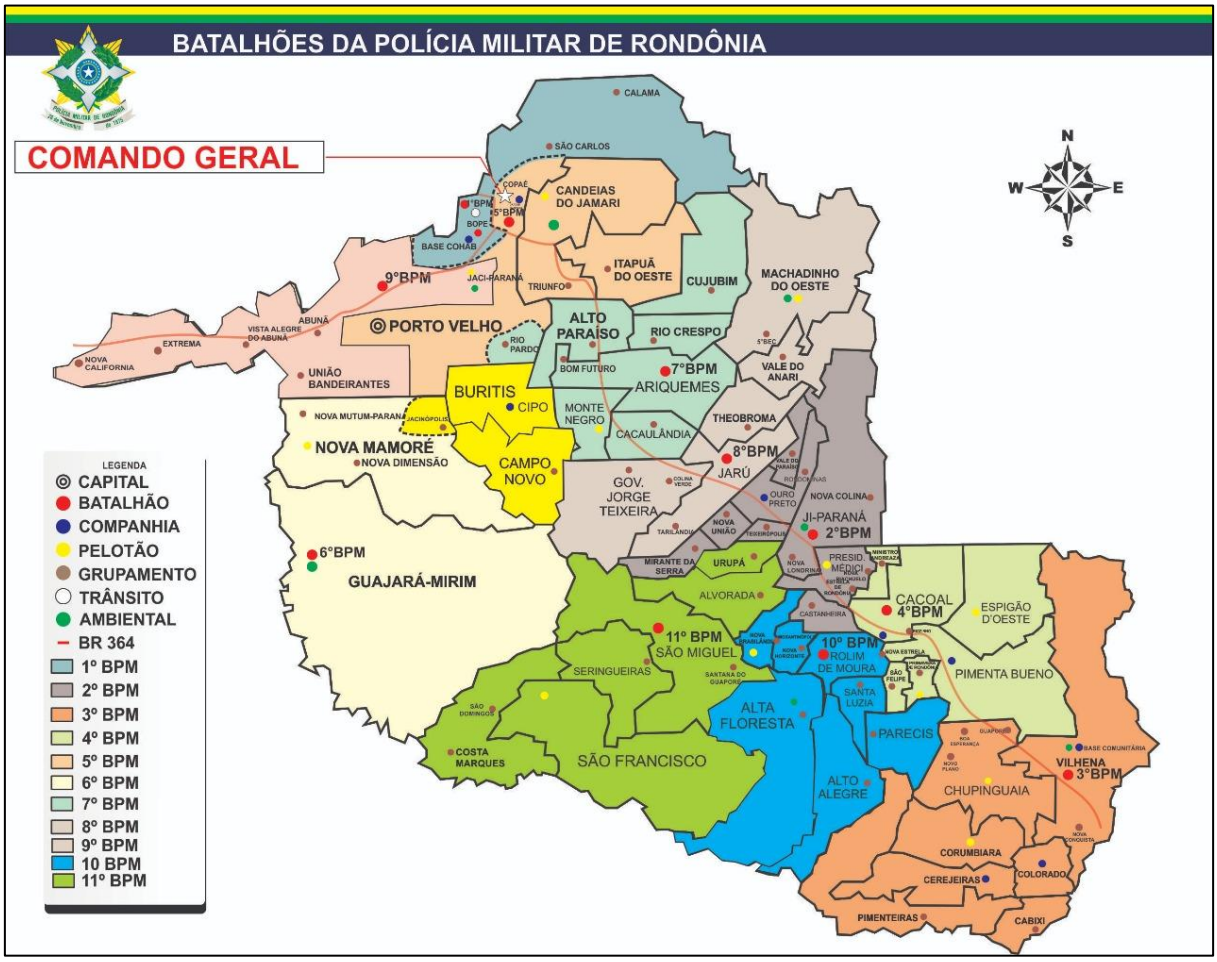
2 REFERENCIAL TEÓRICO

2.1 Polícia Militar do Estado de Rondônia

A Polícia Militar do Estado de Rondônia (à época Território Federal de Rondônia) criada no dia 26 de novembro de 1975 através da Lei Federal nº 6.270 e regulamentada em 11 de janeiro de 1977 pelo Decreto 79.108, pelo então Presidente da República Ernesto Geisel, está presente nos 52 (cinquenta e dois) municípios e distritos do Estado e conta com um efetivo acima de 5.200 policiais militares. Esta força policial de acordo com o art. 144 da Constituição da República Federativa do Brasil, tem como responsabilidade primordial a preservação da ordem pública e a segurança dos cidadãos. A Polícia Militar é uma das instituições responsáveis pela segurança pública, juntamente com os corpos de bombeiros militares, as polícias civis, polícia federal, rodoviária federal, ferroviária federal e polícias penais federais, estaduais e distrital, todas descritas no referido artigo constitucional.

A principal atribuição da Polícia Militar é o policiamento ostensivo, ou seja, ações preventivas e repressivas para manter a tranquilidade, a paz social e o cumprimento das leis. Ela atua na prevenção e repressão de crimes, no controle de distúrbios civis, na proteção do patrimônio público e privado, além de realizar o policiamento de trânsito em algumas jurisdições onde existe convênio firmado com órgãos de trânsito. A Polícia Militar é subordinada ao governador do estado, sendo uma força auxiliar e reserva do Exército Brasileiro. Ela possui uma estrutura hierárquica e disciplinar própria, baseada nos princípios da hierarquia e disciplina militares.

Figura 1 - Distribuição dos Batalhões em Rondônia



Fonte: Acervo da PM-RO (2018)

Todo esse aglomerado organizacional presente em todo estado de Rondônia e até nas áreas mais longínquas, como podemos identificar na Figura 1, carece de ferramentas administrativas para gerir os materiais bélicos empregados no policiamento ordinário, pois esse material requer controle e armazenamento especial, histórico de movimentação e registro de manutenções nos casos em que houver.

2.2 Aspectos Legais

Os Comandantes Gerais, das Polícias Militares estaduais dentre todas suas responsabilidades que possuem, existem uma atribuição específica, a de manter um total controle do material bélico institucional, sendo essa como uma das suas maiores responsabilidades. É possível confirmar tal atribuição no art. 24, § 4º do Decreto nº 9.847/2019:

> “Art. 24. O porte de arma de fogo é deferido aos militares das Forças Armadas, aos policiais federais, estaduais e distritais, civis e militares, aos

> corpos de bombeiros militares e aos policiais da Câmara dos Deputados e do Senado Federal em razão do desempenho de suas funções institucionais.
> [...]
> **§ 4º Atos dos comandantes-gerais das corporações disporão sobre o porte de arma de fogo dos policiais militares e dos bombeiros militares.”.** (grifo nosso)

A legislação básica inicial que começa a tratar sobre esse controle, é advinda da Lei nº 10.826/2003, conhecida por Estatuto do Desarmamento, a qual estabeleceu dois sistemas de registro de controle de arma de fogo, ambos os sistemas têm como objetivo principal garantir a segurança no uso de armas de fogo.

O primeiro deles é o SINARM - Sistema Nacional de Registro de Armas que é responsabilidade da Polícia Federal que tem como objetivo controlar a circulação de armas de fogo em todo o território nacional para população civil e órgãos de segurança pública (não militares), controlando a comercialização e uso, além de registrar informações sobre proprietários e usuários autorizados.

O segundo é o SIGMA - Sistema de Gerenciamento Militar de Armas que o Exército Brasileiro mantém o registro e controle da fabricação, comercialização e uso das armas de uso restrito das forças armadas (Exército, Marinha e Aeronáutica), das polícias militares, dos corpos de bombeiros militares e dos os CAC’s - Caçadores, Atiradores e Colecionadores. Esse controle inclui o registro de todas as informações sobre elas, como número de série, marca, modelo, calibre e a realização de atividades relacionadas, bem como manutenção e inspeções de segurança.

Atualmente o Decreto nº 10.030/2019, que trata sobre o Regulamento de Produtos Controlados de competência do Exército Brasileiro aborda em seu capítulo I: Dos Processos de Controle e Segurança, nos traz a seguinte legislação no artigo 58º:

> “Art. 58. Os processos de controle de PCE são mecanismos operacionais, **automatizados ou não**, que têm a finalidade de:
> I - verificar a conformidade normativa do PCE em relação ao disposto neste Regulamento;
> II - **produzir indicadores** institucionais;
> III - **fornecer informações** para subsidiar a tomada de decisão; e
> IV - permitir a fiscalização efetiva de PCE pelo Comando do Exército.”. (grifo nosso)

Destarte, cada Polícia Militar estadual possuí a responsabilidade de realizar o controle, emprego e uso do armamento de dotação do estado, não se fazendo valer do SIGMA, muito menos do SINARM, pois são sistemas de controle de âmbito federal

de cada órgão correspondente. As Policias Militares devem desenvolver processos próprios de controle e de seguranças citados na legislação acima, e um sistema informatizado próprio traz uma conformidade regulatória do controle desse patrimônio, ajudando a cumprir as regulamentações governamentais e padrões exigidos, garantindo que os bens sejam registrados e relatados seu emprego e cautelamento corretamente.

2.3 Materiais Bélicos

Todo material necessário para o desempenho das atribuições específicas das Polícias Militares nas ações de defesa interna e territorial são considerados materiais bélicos. O material mais comum e utilizado, as armas de fogo que são comercializadas no país devem seguir as exigências da Portaria 7 do Departamento Logístico (D-LOG) do Exército Brasileiro, de 28 de abril de 2006, institui no art. 5º, Inciso IV, a obrigatoriedade da marcação do número de série, como se segue:

> “Art. 5° As armas fabricadas no país deverão apresentar as seguintes marcações:
> I - nome ou marca do fabricante;
> II - nome ou sigla do País;
> III - calibre;
> IV - número de série impresso na armação, no cano e na culatra, quando móvel;
> V - o ano de fabricação quando não estiver incluído no sistema de numeração serial.”.

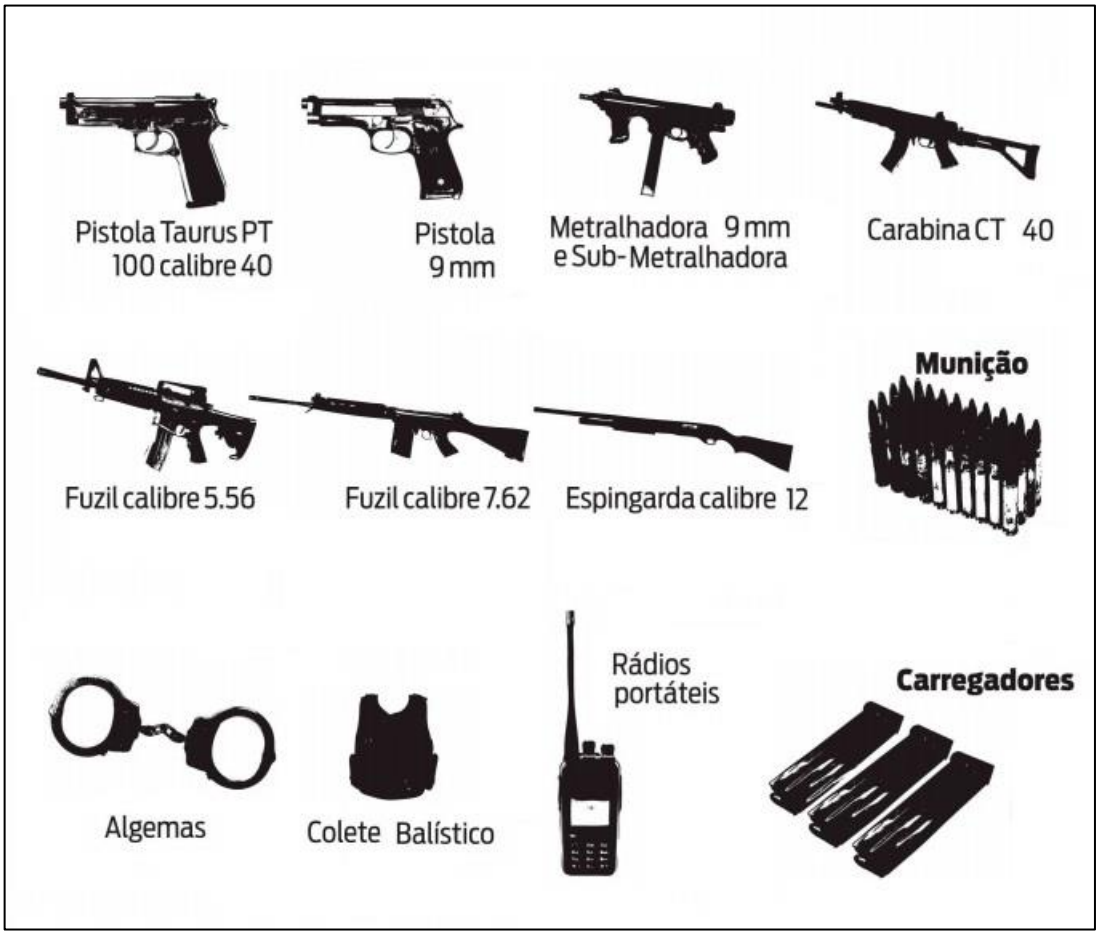
O número de série é o elemento principal identificador do armamento e cada indústria adota o seu critério de padrão da composição do número de série e seus locais de gravação também variam de fabricante para fabricante e até de modelo para modelo do mesmo fabricante. As armas de fogo pertencentes às Polícias Militares em sua maioria apresentam marcas institucionais que podem ser o brasão ou nome da instituição e/ou o número de patrimônio do armamento.

Figura 2 - Pistola PT-100 utilizada pela PMRO

Fonte: Elaborado pelo autor (2023)

Para efeitos de controle institucional os armamentos pertencentes à PMRO são classificados em: espécie (tipo), marca, modelo, calibre, número de série, capacidade de munições e número de patrimônio da arma. Outros materiais utilizados são as metralhadoras e submetralhadoras, fuzis, carabinas, espingardas, algemas, coletes balísticos, rádios portáteis, etc.

Figura 3 - Exemplo de materiais bélicos utilizados pelas Polícias Militares



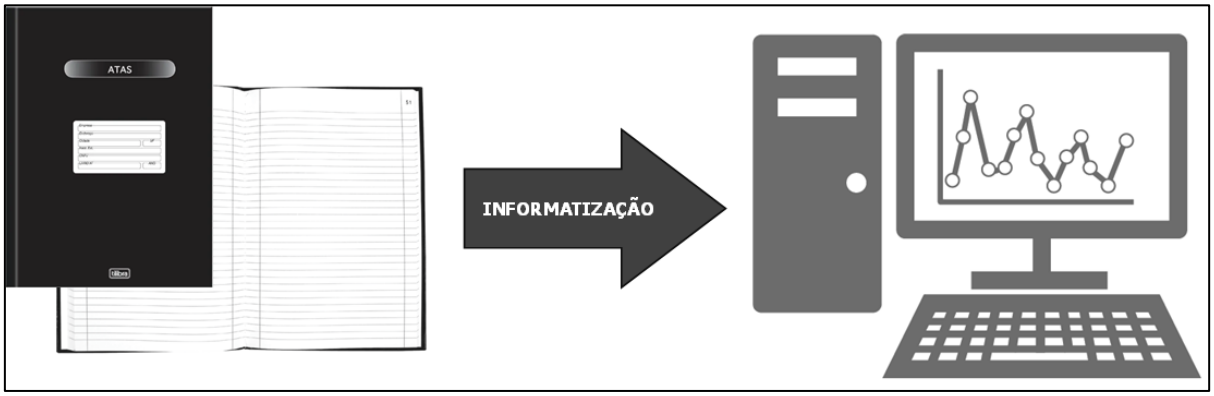
Fonte: elaborado pelo autor (2023) baseado em: https://odia.ig.com.br/noticia/rio-de-janeiro/2015-09-20/armamento-comprado-pelo-estado-vira-objeto-de-destruicao-na-mao-de-criminosos.html

2.4 Informatização do Processo

A utilização de um sistema de informação adequado proporciona uma melhor organização dos dados, facilita o acesso às informações relevantes e contribui para a eficiência e eficácia das atividades realizadas. Pressman (2016) apresenta uma abordagem profissional para a engenharia de *software*, destacando a importância da aplicação de métodos sistemáticos e disciplinados para garantir que a qualidade, a eficiência e a confiabilidade dos produtos de *software*, que quando bem aplicados entregam um sistema de sucesso. Diante do problema exposto, a informatização do processo de controle do material bélico poderá trazer benefícios imensuráveis para atender toda e qualquer complexidade de gestão administrativa.

A elaboração desse sistema de informação será um elemento fundamental para uma melhor gestão, pois a sua finalidade é tratar, armazenar e fornecer dados e informações com a emissão de relatórios que vão auxiliar em um melhor gerenciamento e nos casos que forem necessários a tomada de decisões. Com a informatização, será possível centralizar e consolidar os dados relacionados ao controle de material bélico, evitando redundâncias e inconsistências. O acesso às informações será facilitado, proporcionando maior agilidade e precisão na consulta e análise dos dados.

Figura 4 - Informatização de um processo manual



Fonte: Elaborado pelo autor (2023)

Em suma, a informatização de um processo manual, como o controle de material bélico, é essencial para otimizar a gestão administrativa, garantir a qualidade e segurança das informações, reduzir o tempo de execução e auxiliar na tomada de decisões.

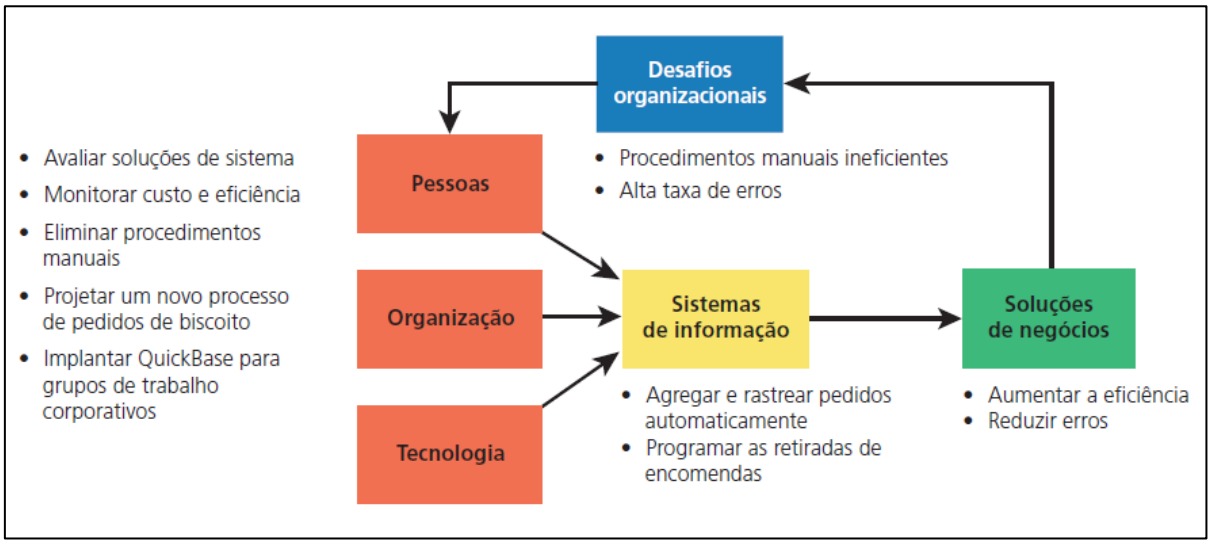
2.4.1 Sistemas de Informação Gerenciais

O objetivo de um sistema de informação compreende em um *software* que tem interação entre procedimentos, pessoas e tecnologias que trabalham em conjunto coletando, processando, armazenando e disseminando informações para dar suporte às funções de planejamento, controle e organização de determinado processo ou negócio. Atualmente existem alguns tipos de sistemas de informação, onde cada um possui especificidades e particularidades, porém diversos sistemas trabalham de maneira integrada, podendo variar em níveis operacionais, de conhecimento, estratégicos e táticos.

O projeto contempla a elaboração do sistema baseada na literatura dos Sistemas de Informações Gerenciais - SIG, pois ele é voltado para geração de produtos de informação que apoiam as tomadas de decisão administrativas que é o objetivo desse projeto: gerenciar o patrimônio bélico de uma OPM. Através de um SIG aplicado na área operacional alcançaremos um controle mais rigoroso da qualidade dos serviços e na área administrativa será promovido uma otimização do controle e da gestão do patrimônio. Como são gerados relatórios a partir do banco de dados utilizado, o fornecimento de informações ajuda o gestor com para uma melhor gestão, controle e tomada de decisões.

Os Sistemas de Informações Gerenciais são projetados para atender às necessidades específicas de cada organização, adaptando-se aos processos e fluxos de trabalho existentes. Conforme mencionado por Laudon (2014), esses sistemas podem ser personalizados de acordo com os requisitos de informação da empresa, permitindo a coleta e análise de dados relevantes para a tomada de decisões. A flexibilidade dos SIG é um fator importante, pois cada organização possui características e necessidades únicas, exigindo soluções personalizadas para aprimorar a eficiência e eficácia dos processos.

Figura 5 - Diagrama de desenvolvimento de um sistema de informação



Fonte: LAUDON (2014) p.403

Outra vantagem dos Sistemas de Informações Gerenciais é a capacidade de integração com outras tecnologias e sistemas existentes. Laudon (2014) destacam que os SIG podem ser integrados a outros sistemas empresariais, como sistemas de

gestão de recursos humanos, sistemas de contabilidade e sistemas de gerenciamento de clientes. Essa integração permite a troca de informações entre os diferentes sistemas, melhorando a eficiência e a precisão dos processos organizacionais.

Além disso, os SIG auxiliam na criação de uma cultura de dados e informação dentro da organização. De acordo com Laudon (2014), a implementação de um SIG promove a coleta sistemática de dados, o que incentiva a cultura de análise e tomada de decisões baseadas em informações concretas. Os gestores passam a contar com relatórios, análises e indicadores que fornecem *insights* valiosos sobre o desempenho da organização, permitindo a identificação de tendências, a detecção de problemas e a busca por oportunidades de melhoria contínua.

Normalmente existe uma dificuldade de avaliar, de forma quantitativa, qual o benefício essencial de um sistema de informação gerencial, ou seja, qual a melhoria que ele trará. Segundo Oliveira (2018) existe uma lista de possíveis hipóteses sobre os impactos que os sistemas gerenciais trazem, ainda que de forma genérica podemos citar:

- Redução de custo das operações;
- Melhoria de acesso às informações, propiciando relatórios mais precisos e rápidos, com menor esforço;
- Melhoria na produtividade;
- Mais segurança de acesso à informação;
- Melhoria na tomada de decisão, através das informações geradas;
- Aumento do nível de motivação e comprometimento das pessoas envolvidas.

2.4.2 PHP

Segundo o manual oficial da linguagem, ele define que o PHP é “uma linguagem de *script open source* de uso geral, muito utilizada e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do HTML”. (THE PHP GROUP, 2022)

Os motivos da escolha dessa linguagem para execução desse projeto são por ela possuir uma ampla adoção de uso e tendo como consequência uma comunidade

ativa, por ser uma das linguagens mais populares para o desenvolvimento *web*. Isso significa que há uma abundância de recursos, documentação e suporte disponíveis. O reflexo desta abundância de recursos e tutoriais disponíveis torna o desenvolvimento em PHP mais acessível e rápido, permitindo que desenvolvedores construam sistemas de forma eficiente.

Esta linguagem possui suporte nativo para uma ampla variedade de bancos de dados, como MySQL, PostgreSQL e Oracle. Isso torna mais fácil e conveniente a integração de um sistema PHP com um banco de dados, permitindo a manipulação eficiente e segura dos dados. E a sua flexibilidade permite que os desenvolvedores construam sistemas simples ou complexos de acordo com as necessidades do projeto. Além disso, existem muitos *frameworks* e bibliotecas disponíveis para PHP, como Laravel, Symfony e CodeIgniter, que fornecem estruturas sólidas para desenvolver aplicações robustas e escaláveis. Além do mais a sua integração ao HTML com o CSS (*Cascading Style Sheets)* e JavaScript permite a criação de páginas dinâmicas e interativas, melhorando a experiência do usuário.

2.4.3 HTML

O Hyper Text Markup Language é uma linguagem que apresenta as informações na internet, tem uma importante função em aplicações *web* o navegador utilizado interpreta o código HTML e apresenta quando uma página é aberta. É ele que tem a função de fornecer a interface básica para a interação com o PHP, além disso também é através dela que apresentaremos as interfaces gráficas para o usuário final utilizador do sistema (W3SCHOOLS, 2022).

A linguagem tem como função também de estruturar o conteúdo e apresentação de páginas *web*. De acordo com Pilgrim (2012), o HTML é uma linguagem baseada em marcações, onde elementos são utilizados para definir a estrutura e organização do conteúdo em uma página *web*. Enfatiza a importância de seguir as práticas recomendadas e as diretrizes do HTML para garantir uma melhor compatibilidade entre diferentes navegadores e dispositivos. Ele fornece exemplos claros e explicativos para ilustrar o uso adequado das *tags* e elementos HTML, facilitando o entendimento e a aplicação prática dos conceitos abordados. Através do

HTML, é possível definir os diferentes elementos presentes em uma página, como títulos, parágrafos, listas, imagens, links e tabelas.

A linguagem utiliza *tags*, que são inseridas no código HTML, para delimitar e identificar cada elemento. Por exemplo, a *tag* "<p>" é utilizada para criar um parágrafo, enquanto a *tag* "<img>" é usada para inserir uma imagem. Além disso, o HTML permite a criação de links entre páginas, possibilitando a navegação e interconexão entre diferentes documentos *web*. Também é possível incorporar conteúdo externos, como vídeos e áudios, através de elementos específicos do HTML.

O HTML é uma linguagem de fácil aprendizado e amplamente utilizada na construção de páginas *web*. É interpretada pelos navegadores, que renderizam o código HTML e exibem o conteúdo da página conforme as marcações e elementos definidos. No entanto, o HTML por si só não é capaz de adicionar comportamentos ou interatividade avançada às páginas *web*, sendo comum a combinação com outras linguagens, como o CSS (Cascading Style Sheets) e o JavaScript, para aprimorar a apresentação e adicionar funcionalidades dinâmicas. (PILGRIM, 2012)

2.4.4 UML

A Unified Modeling Language é uma linguagem gráfica amplamente utilizada na engenharia de *software* para visualizar, projetar, especificar e documentar sistemas complexos. Ela fornece um conjunto de diagramas que ajudam a descrever diferentes aspectos de um sistema, desde a estrutura estática até o comportamento dinâmico.

A UML permite que os desenvolvedores comuniquem e compartilhem informações sobre um sistema de forma padronizada e compreensível. Ela utiliza diversos tipos de diagramas para representar diferentes perspectivas do sistema. Alguns dos diagramas mais comuns incluem:

- Diagrama de Classes: Descreve a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas.
- Diagrama de Sequência: Ilustra a interação entre objetos ao longo do tempo, representando a ordem das mensagens trocadas entre os objetos.
- Diagrama de Casos de Uso: Descreve a interação entre atores externos e o sistema, mostrando os diferentes casos de uso e seus relacionamentos.

- Diagrama de Atividades: Modela o fluxo de atividades do sistema, mostrando a sequência de ações, decisões, ramificações e paralelismo.
- Diagrama de Componentes: Representa os componentes físicos ou lógicos do sistema e as dependências entre eles.

Segundo Guedes (2010) a UML é uma linguagem visual que pode ser facilmente compreendida por diferentes membros de uma equipe de desenvolvimento de *software*, como analistas, *designers* e programadores. Ela fornece uma representação abstrata do sistema, permitindo que os desenvolvedores se concentrem nas principais características e funcionalidades.

A utilização da UML traz diversos benefícios, como a melhoria na comunicação entre os membros da equipe, a detecção antecipada de problemas de *design* e a documentação clara e consistente do sistema. Além disso, a UML permite a geração automática de código a partir dos diagramas, acelerando o processo de desenvolvimento.

2.4.5 SQL

Para Silberschatz et al. (2011) a Structured Query Language é uma linguagem de programação projetada para gerenciar e manipular bancos de dados relacionais. Ela fornece um conjunto de comandos e instruções que permitem a criação, consulta, modificação e exclusão de dados em um banco de dados.

A linguagem SQL é amplamente utilizada em Sistemas de Gerenciamento de Bancos de Dados Relacionais - SGBDR, como MySQL, Oracle, SQL Server, PostgreSQL, entre outros. Ela permite que os desenvolvedores acessem e interajam com os dados armazenados em um banco de dados de forma eficiente e estruturada.

Existem diferentes categorias de comandos SQL, incluindo:

- Comandos de DDL (Data Definition Language): São utilizados para definir a estrutura do banco de dados, como a criação de tabelas, definição de colunas, índices e restrições.

- Comandos de DML (Data Manipulation Language): São utilizados para manipular os dados no banco de dados, como inserção (INSERT), atualização (UPDATE) e exclusão (DELETE) de registros em uma tabela.
- Comandos de Consulta (SELECT): São utilizados para recuperar dados de uma ou mais tabelas do banco de dados. Esses comandos permitem filtrar, ordenar e agrupar os resultados, além de realizar operações de junção entre tabelas.

A linguagem SQL oferece uma sintaxe padronizada e uma abordagem declarativa, o que significa que o desenvolvedor especifica o que deseja obter e não precisa se preocupar com os detalhes de como os dados serão recuperados ou manipulados pelo banco de dados.

A utilização da linguagem SQL traz diversos benefícios. Ela permite que os desenvolvedores realizem consultas complexas e eficientes em bancos de dados, recuperando apenas os dados necessários para uma determinada tarefa. Além disso, a SQL é uma linguagem altamente flexível e portátil, pois os comandos podem ser executados em diferentes sistemas de gerenciamento de bancos de dados relacionais (SILBERSCHATZ, 2011).

3 MATERIAIS E MÉTODOS

Este capítulo apresenta as ferramentas de desenvolvimento utilizadas para elaboração do sistema *web*. Também serão apresentadas as metodologias e requisitos para atingir o resultado final.

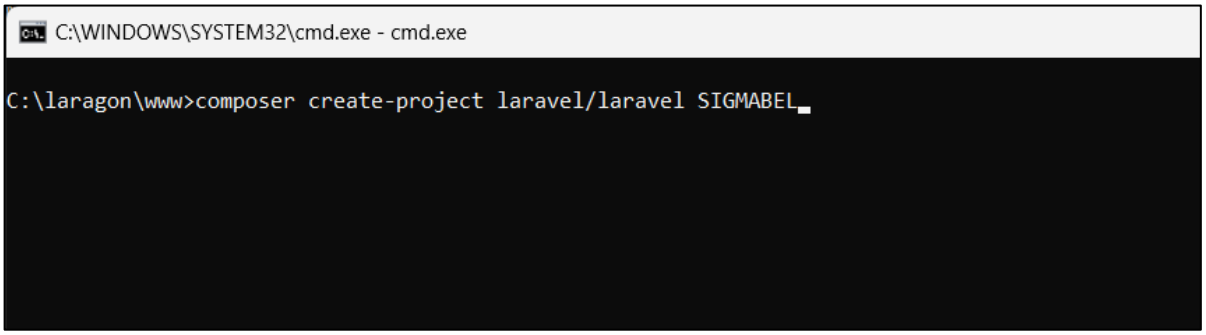
3.1 Materiais

Existem diversas ferramentas de desenvolvimento disponíveis para criar sistemas *web* em PHP, oferecendo recursos e facilidades que podem acelerar o processo de desenvolvimento. Algumas das ferramentas mais populares fornecem recursos avançados de edição de código, depuração e gerenciamento de projetos. Além disso, existem *frameworks* que fornecem uma estrutura sólida e organizada para o desenvolvimento de aplicativos PHP, facilitando a criação de rotas, modelos de dados, autenticação e outras funcionalidades comuns. Essas ferramentas podem ser altamente eficientes e produtivas, permitindo que os desenvolvedores se concentrem no desenvolvimento de recursos e na lógica do aplicativo, ao invés de se preocuparem com detalhes de baixo nível. Para elaboração deste sistema foram utilizadas as ferramentas descritas abaixo.

3.1.1 Laravel 10

O Laravel é um *framework* de desenvolvimento de *software web* em PHP livre e gratuito, criado em 2011 pelo programador canadense Taylor Otwell para facilitar a construção de aplicações *web* robustas, seguras e escaláveis. O *framework* possui uma arquitetura MVC (*Model-View-Controller*) que separa as camadas de lógica de negócio, apresentação e controle, permitindo uma melhor organização do código e facilitando a manutenção da aplicação. A sua versão mais recente é a 10, lançada em 2023, que traz diversas melhorias e novas funcionalidades. O Laravel se destaca pela sua simplicidade e elegância, além de ter uma ampla documentação e uma comunidade de desenvolvedores bastante ativa de desenvolvedores que criam e compartilham pacotes e extensões para o *framework*.

Figura 6 - Comando de criação de um projeto Laravel



Fonte: elaborado pelo autor (2023)

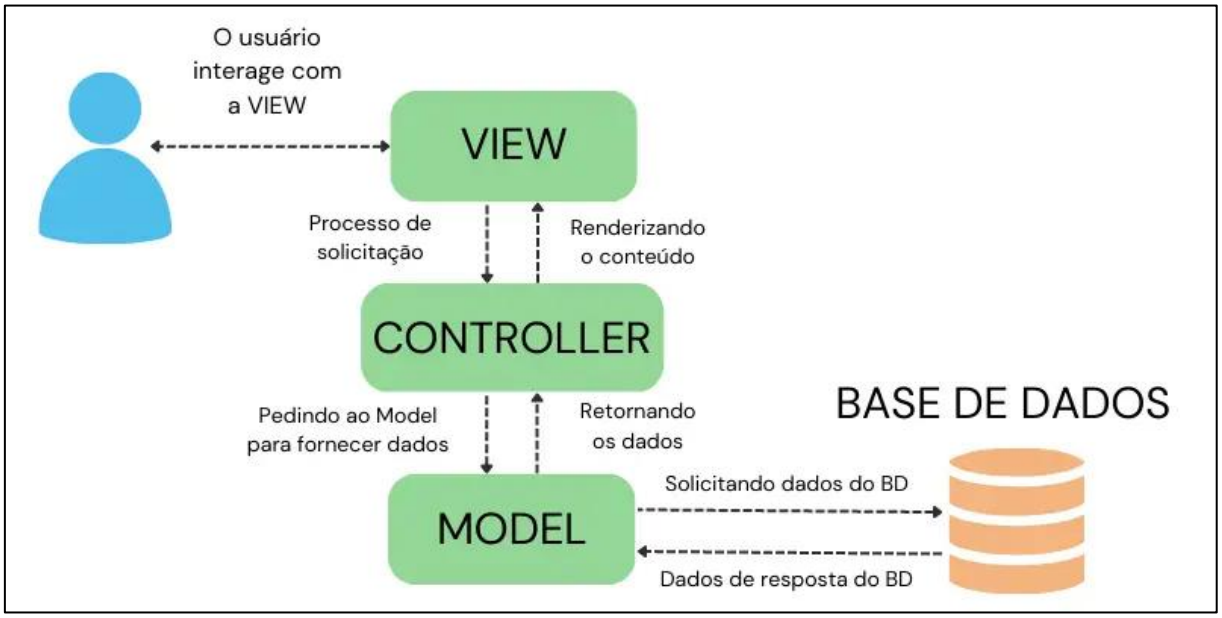
Com o Laravel 10, é possível desenvolver aplicações *web* de maneira mais rápida e eficiente, graças a seus recursos, que facilita a compilação e otimização de arquivos CSS e JavaScript, e o Laravel Horizon, que permite monitorar e gerenciar filas de processamento de forma intuitiva e visual. Além disso, o Laravel oferece uma série de recursos e funcionalidades que simplificam tarefas comuns no desenvolvimento *web*, como autenticação de usuários, gerenciamento de sessões, validação de formulários, envio de e-mails, cache, entre outros. O *framework* também traz uma série de melhorias de desempenho e segurança, tornando-o uma opção ainda mais interessante para projetos de desenvolvimento *web* que buscam uma solução completa e eficiente para seu projeto.

3.1.1.1 MVC - Model-View-Controller

O padrão de arquitetura MVC - *Model-View-Controller* é amplamente utilizado no desenvolvimento de aplicativos da *web*, incluindo no contexto do PHP. O MVC divide uma aplicação em três componentes principais: o *Model* (modelo), o *View* (visão) e o *Controller* (controlador). Essa abordagem organiza o código de uma maneira que separa as responsabilidades e facilita a manutenção e escalabilidade do sistema.

Segundo Fowler (2007) o modelo (*Model*) representa a camada de dados da aplicação. Ele lida com a lógica de negócios, acesso a banco de dados e manipulação dos dados subjacentes. O modelo é responsável por recuperar, atualizar, excluir e criar registros de dados, bem como realizar validações e cálculos necessários.

Figura 7 - Processo do MVC



Fonte: https://www.usandopy.com/media/uploads/2023/03/22/pt-cover-website-22.png

A visão (*View*) é responsável pela apresentação dos dados ao usuário. Ela exibe as informações processadas pelo modelo de uma maneira compreensível e agradável. A visão pode ser uma página HTML, um *template* ou qualquer outra forma de apresentação de informações para o usuário. É importante ressaltar que a visão deve ser passiva e não deve conter lógica de negócios complexa (FOWLER, 2007).

O controlador (*Controller*) atua como intermediário entre o modelo e a visão. Ele recebe as solicitações do usuário, interage com o modelo para buscar ou atualizar dados e, em seguida, envia esses dados à visão correta para renderização e exibição. O controlador é responsável por coordenar as ações do usuário e manipular eventos relacionados à interação (FOWLER, 2007).

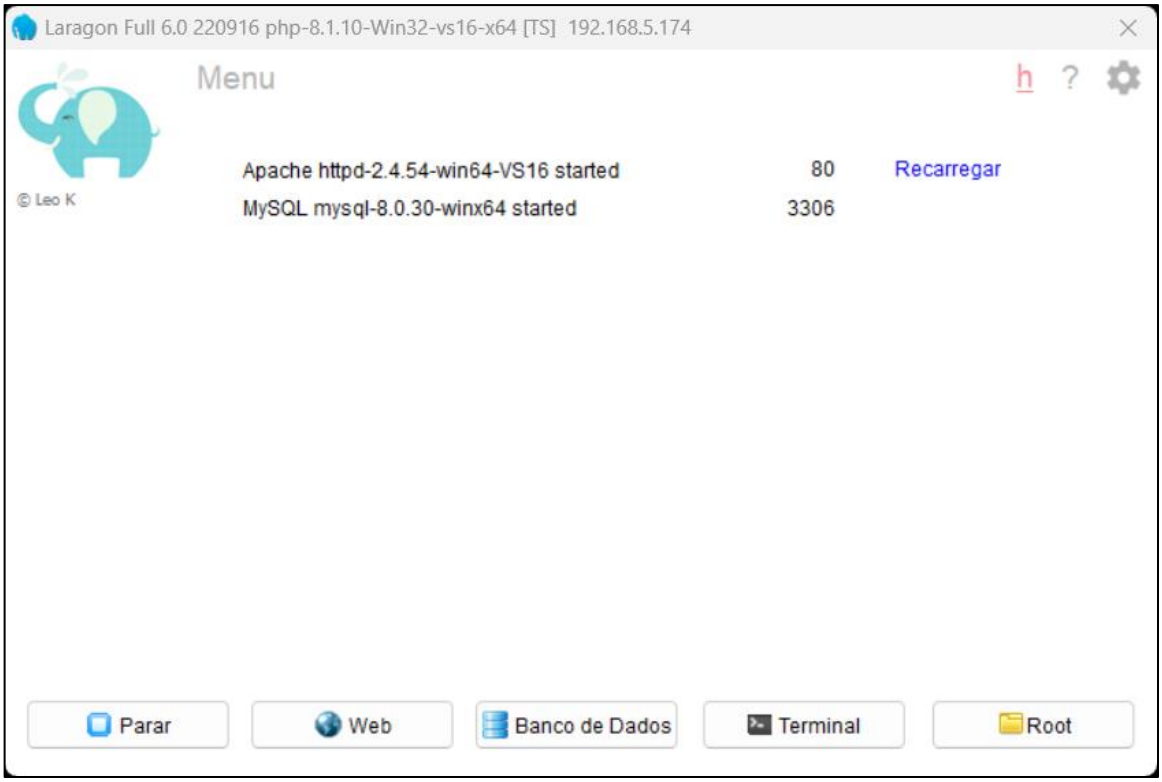
Essa separação de responsabilidades entre modelo, visão e controlador permite que cada componente seja desenvolvido e mantido independentemente. Alterações no modelo não afetam diretamente a visão e vice-versa, facilitando a manutenção e o teste de cada componente isoladamente. Além disso, o MVC promove a reutilização de código, pois diferentes visões podem ser construídas para interagir com o mesmo modelo e controlador (FOWLER, 2007).

3.1.2 Laragon

Conforme descrito pelo desenvolvedor da ferramenta em seu site oficial o Laragon é “um ambiente de desenvolvimento universal portátil, isolado, rápido e poderoso para PHP, Node.js, Python, Java, Go e Ruby. É rápido, leve, fácil de usar e fácil de entender”. E ainda completa que ele é “ótimo para construir e gerenciar aplicações *web* modernas. Ele é focado no desempenho e projetado em torno da estabilidade, simplicidade, flexibilidade e liberdade” (LARAGON, 2022).

O Laragon é o ambiente de desenvolvimento *web* que atende todos os requisitos citados pois nos permite ter um servidor *web* pronto com menor esforço, é um ambiente isolado que não gera conflitos e possui um virtual *host* embutido, com uma URL própria auxiliando o programador lhe dando mais tempo para programar sem se importar com a configuração das ferramentas. Ele oferece uma interface gráfica do usuário para gerenciar e configurar esses serviços, permitindo que os desenvolvedores mudem facilmente a versão do PHP, alterem as configurações do servidor *web* e do banco de dados, instalem bibliotecas adicionais, tornando mais fácil para os desenvolvedores criar e executar projetos baseados nesses *frameworks*.

Figura 8 - Visão geral do programa Laragon



Fonte: elaborado pelo autor (2023)

Em resumo, o Laragon é uma ferramenta de desenvolvimento *web* poderosa e fácil de usar para Windows, projetada para tornar mais fácil para os desenvolvedores trabalharem em seus projetos *web* locais.

3.1.2.1 Apache

O servidor HTTP Apache é um dos servidores *web* mais populares e amplamente utilizados. Ele é um *software* de código aberto e gratuito, desenvolvido e mantido pela Apache Software Foundation. O Apache é conhecido por sua confiabilidade, desempenho e flexibilidade, tornando-o uma escolha popular para hospedar sites e aplicações *web*. O Apache é compatível com vários sistemas operacionais, como Windows, Linux, macOS e outros. Ele suporta uma ampla gama de recursos, incluindo suporte a linguagens de programação como PHP, Python, Perl, entre outras. Além disso, o Apache também oferece recursos avançados de segurança, permitindo a configuração de autenticação, criptografia SSL/TLS - (Secure Sockets Layer/ Transport Layer Security) e controle de acesso a diretórios e arquivos.

No contexto de projetos WAMP (Windows, Apache, MySQL e PHP), o Apache é um componente essencial para criar um ambiente de desenvolvimento local para desenvolvimento de sites. O WAMP é uma combinação de *software* livre que permite que o desenvolvedor execute aplicativos *web* em um ambiente de servidor local antes de implantá-los em um servidor de hospedagem. O Apache é responsável por fornecer o servidor *web* no ambiente local, permitindo que os desenvolvedores testem seus sites e aplicativos em um ambiente controlado antes de disponibilizá-los publicamente.

Ao usar o Apache em projetos locais, os desenvolvedores podem configurar facilmente um servidor local no qual podem desenvolver, testar e depurar seus sites. Eles podem executar código PHP, acessar bancos de dados MySQL e realizar todas as operações necessárias para desenvolver um site funcional. O Apache é altamente configurável, permitindo que os desenvolvedores personalizem as configurações do servidor de acordo com as necessidades de seus projetos.

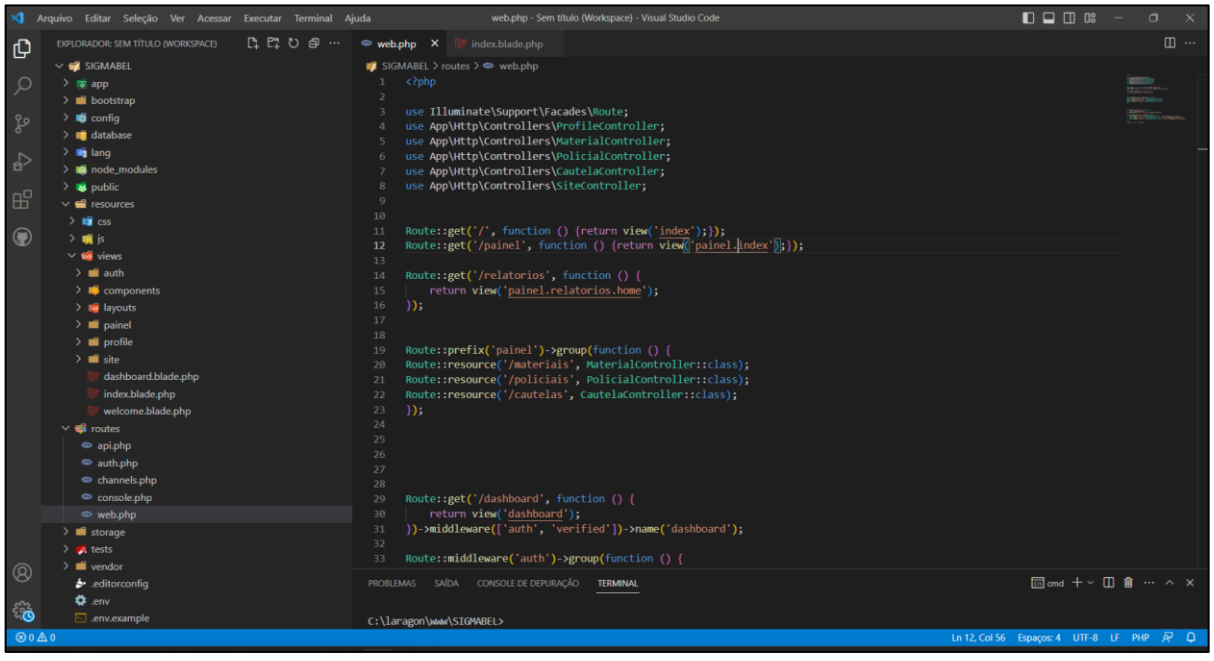
3.1.3 Visual Studio Code

Também conhecido como VSCode, é uma ferramenta de ambiente de desenvolvimento integrado (IDE) desenvolvido pela Microsoft, altamente

personalizável e de código aberto. Escolhido para ser a ferramenta de edição das linguagens de programação deste projeto. Conforme está descrito no site oficial do Visual Studio Code (2022), ele é “um editor de código-fonte leve e executado em áreas de trabalho, está disponível para Windows, macOS e Linux, possui um rico ecossistema de extensões para várias linguagens” em especial para o PHP, a linguagem escolhida onde conta também com extensões de conexão com os bancos de dados mais comuns do meio da programação *web*.

Ele oferece uma ampla gama de recursos e funcionalidades para facilitar o processo de desenvolvimento de *software*. Ele suporta várias linguagens de programação, incluindo JavaScript, Python, C#, HTML, CSS e muitas outras. Com seu editor de código avançado, fornece recursos como realce de sintaxe, formatação automática, sugestões inteligentes de código e navegação eficiente dentro dos arquivos do projeto.

Figura 9 - Visão geral do programa Visual Studio Code



Fonte: elaborado pelo autor (2023)

Uma das características marcantes do Visual Studio Code é sua extensibilidade. Ele permite aos desenvolvedores adicionar e utilizar uma variedade de extensões disponíveis no mercado. Essas extensões oferecem funcionalidades extras, como suporte a *frameworks* específicos, integração com ferramentas de controle de versão, depuração avançada, *snippets* de código e muito mais. Isso

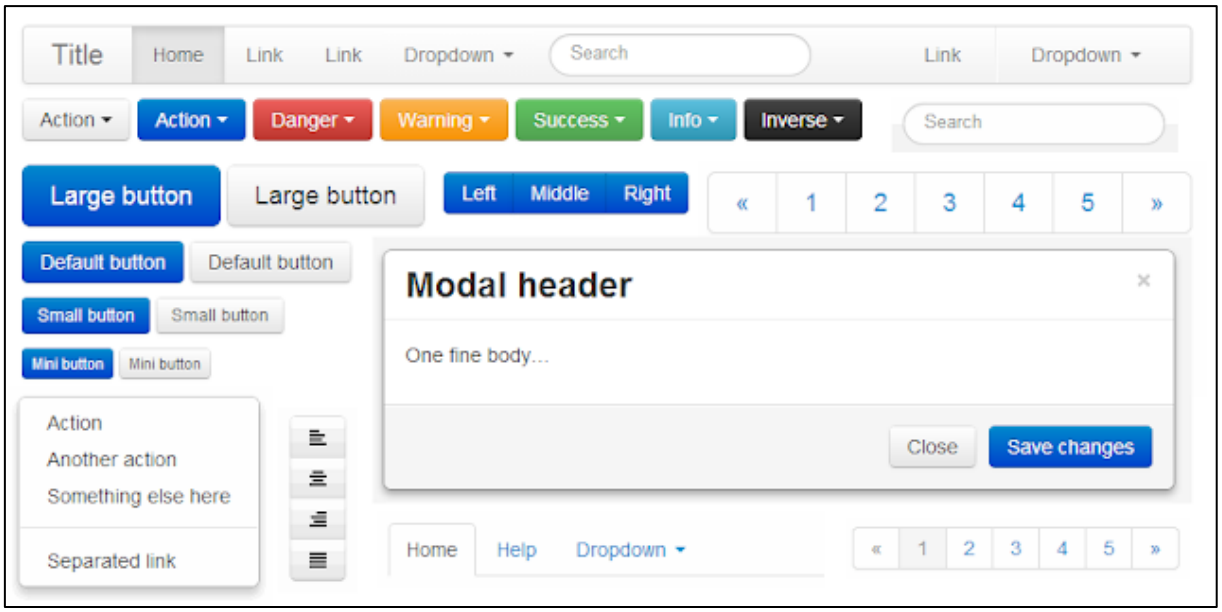
permite personalizar a experiência de desenvolvimento de acordo com as preferências e necessidades individuais.

Outro ponto forte do Visual Studio Code é seu sistema de depuração integrado. Os desenvolvedores podem depurar seus programas por meio de recursos como pontos de interrupção, inspeção de variáveis, rastreamento de pilha e console interativo. Isso ajuda a identificar e corrigir erros e problemas de forma eficiente durante o processo de desenvolvimento.

3.1.4 Bootstrap

O Bootstrap é um *framework front-end* de código aberto muito utilizado na programação em PHP e outras linguagens *web*. Ele foi desenvolvido por *designers* do Twitter em meados de 2010 e tem como objetivo facilitar a criação de interfaces responsivas e estilizadas para páginas *web*. De acordo com o site oficial do Bootstrap (2023), ele oferece um conjunto de estilos CSS e componentes JavaScript pré-construídos, que podem ser utilizados para agilizar o desenvolvimento e garantir uma aparência visualmente atraente e consistente em diferentes dispositivos e tamanhos de tela.

Figura 10 - Componentes de interface do *framework* Bootstrap



Fonte: http://www.d3noob.org/2013/07/getting-started-using-bootstrap-with.html

Com o Bootstrap, os desenvolvedores em PHP podem aproveitar uma ampla gama de recursos e funcionalidades prontas para uso. O *framework* fornece uma

estrutura de *grid* responsiva, que permite organizar o *layout* da página de forma flexível, adaptando-se automaticamente ao tamanho da tela do dispositivo em que está sendo visualizada. Além disso, o Bootstrap oferece uma biblioteca de componentes, como botões, formulários, navegação, alertas e muito mais, que podem ser facilmente incorporados às páginas PHP, proporcionando uma experiência interativa e amigável para os usuários.

Uma das vantagens do uso do Bootstrap em conjunto com o PHP é a economia de tempo e esforço. Com seus estilos e componentes pré-definidos, o Bootstrap elimina a necessidade de criar tudo do zero, permitindo que os desenvolvedores se concentrem na lógica e funcionalidade do aplicativo PHP. Além disso, o Bootstrap é altamente personalizável, permitindo que os desenvolvedores modifiquem os estilos e comportamentos dos componentes de acordo com as necessidades específicas do projeto em PHP.

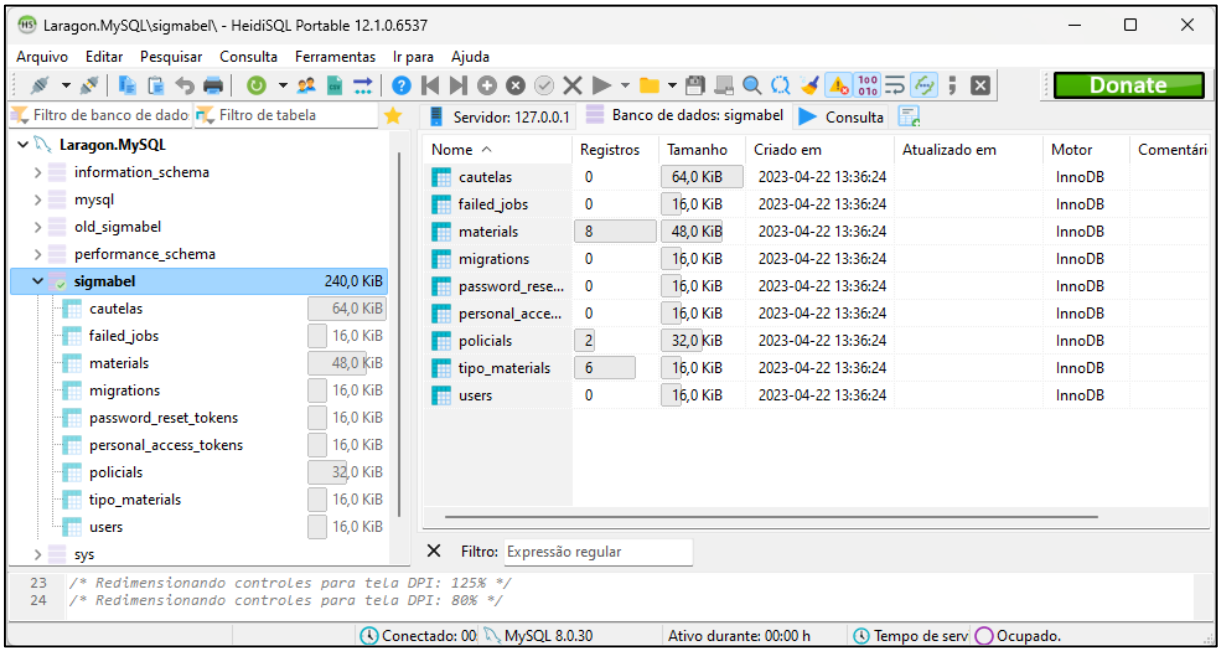
Em suma, o Bootstrap é um *framework* popular na programação em PHP que oferece um conjunto de estilos e componentes pré-construídos, facilitando o desenvolvimento de interfaces responsivas e estilizadas para páginas *web*. Sua utilização traz benefícios como a economia de tempo, a consistência visual e a adaptabilidade a diferentes dispositivos.

3.1.5 HeidSQL

O HeidiSQL é uma ferramenta de *software* livre utilizada para gerenciar banco de dados, possui integração com os melhores e mais utilizados em bancos de dados disponíveis para versão *web*, como o MariaDB e MySQL. É o escolhido para o projeto por possuir uma janela simples e intuitiva onde é possível criar e editar as tabelas e manipular os dados auxiliando o desenvolvedor a visualização das interações CRUD do sistema com o banco (HEIDISQL, 2022).

Com o HeidiSQL, os usuários podem realizar uma ampla gama de tarefas de gerenciamento de banco de dados, como criar, modificar e excluir bancos de dados, tabelas e campos, além de executar consultas SQL, importar e exportar dados, gerenciar usuários e privilégios, entre outras funcionalidades.

Figura 11 - Visão geral do programa HeidiSQL



Fonte: elaborado pelo autor (2023)

A interface do programa é organizada de forma clara, com painéis que exibem informações sobre a estrutura do banco de dados, as consultas executadas e os resultados obtidos. Os usuários podem navegar facilmente pelos objetos do banco de dados, visualizar e editar dados nas tabelas, e executar consultas complexas para recuperar informações específicas.

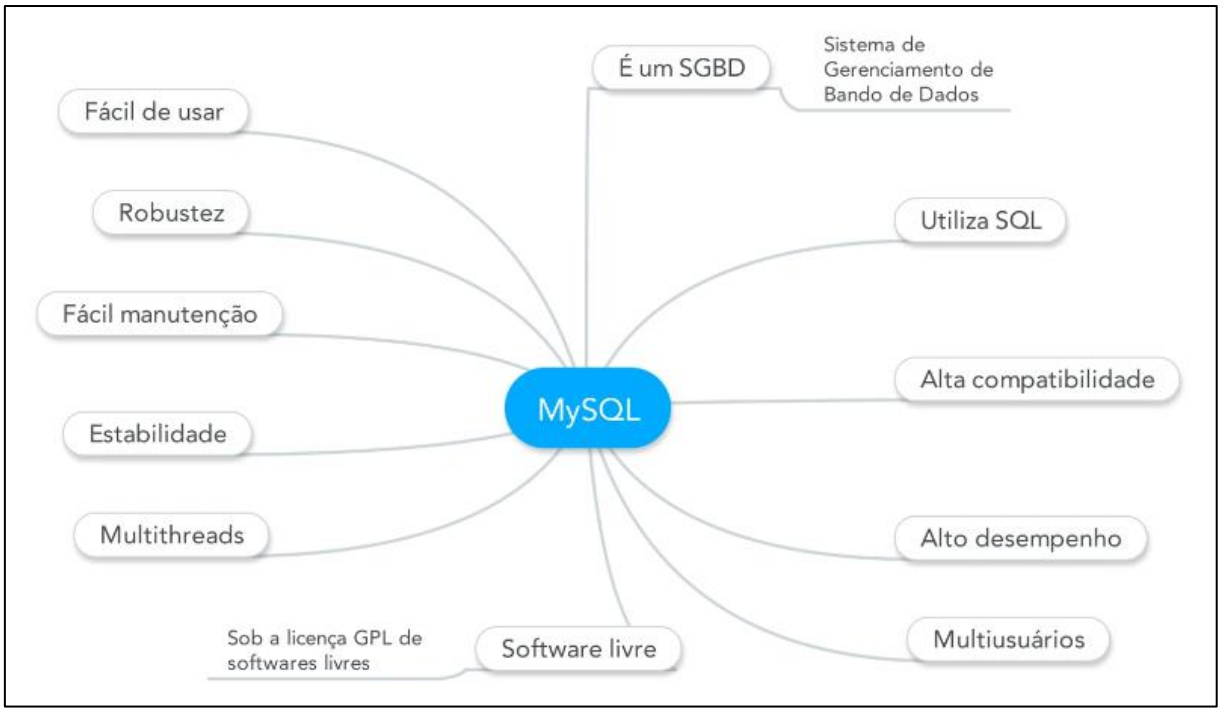
Além disso, o HeidiSQL oferece recursos avançados, como a capacidade de salvar consultas frequentes como favoritos, agendar *backups* automáticos, importar e exportar dados em diferentes formatos (como CSV, SQL, XML), e estabelecer conexões seguras por meio do protocolo SSH.

3.1.5.1 MySQL

Segundo Bento (2018) o MySQL é um Sistema de Gerenciamento de Banco de Dados Relacional - SGBDR de código aberto amplamente utilizado em aplicações *web* e empresariais. Ele fornece um ambiente seguro, escalável e confiável para armazenar e gerenciar grandes volumes de dados de forma eficiente. Ele permite a criação e organização de bancos de dados compostos por tabelas, que são estruturadas em linhas e colunas. Cada tabela contém registros individuais, representando entidades ou informações específicas. Os dados são armazenados de

forma organizada, permitindo a recuperação rápida e eficiente das informações necessárias.

Figura 12 - Mapa mental das vantagens do MySQL



Fonte: CARVALHO (2015) p. 2, fig. 1.1

O MySQL também oferece suporte a recursos avançados, como chaves primárias, índices, restrições de integridade, *triggers* e procedimentos armazenados. Esses recursos permitem a definição de regras de integridade de dados, execução de ações automáticas em resposta a eventos e a criação de funcionalidades mais complexas diretamente no banco de dados. Além do mais ele possui uma comunidade ativa de desenvolvedores e oferece suporte abrangente por meio de sua documentação oficial, fóruns de discussão e recursos online. Isso permite que os usuários encontrem orientações detalhadas, exemplos práticos e soluções para desafios específicos durante o desenvolvimento e administração de bancos de dados MySQL.

3.1.6 Google Chrome

O Google Chrome é um navegador da *web* desenvolvido pelo Google e se destaca por sua velocidade, estabilidade e recursos avançados, tornando-o uma escolha popular para navegação na internet e desenvolvimento de aplicativos *web*. Ele é baseado no projeto de código aberto Chromium e utiliza o motor de renderização

Blink para exibir páginas da *web*. Ele oferece suporte a tecnologias *web* modernas, como HTML5, CSS3 e JavaScript, permitindo que a criação de sites e aplicativos *web* interativos e ricos em recursos sejam exibidos em suas telas.

Para o desenvolvimento *web*, o Google Chrome é uma ferramenta valiosa devido às suas diversas funcionalidades voltadas para programação. Ele possui um conjunto robusto de ferramentas de desenvolvimento integradas, conhecidas como "DevTools", que permitem inspecionar, depurar e otimizar o código HTML, CSS e JavaScript de um site. As ferramentas de desenvolvimento do Google Chrome incluem recursos como o *Console*, que exibe mensagens e erros do JavaScript em tempo real, facilitando a depuração de problemas no código. Além disso, o Chrome oferece recursos para editar em tempo real o código fonte, testar diferentes estilos e *layouts*, monitorar o desempenho do site e simular diferentes dispositivos e resoluções para garantir a responsividade.

3.1.7 Draw.io

O Draw.io é uma ferramenta de diagramação online que permite criar diagramas e fluxogramas de forma intuitiva e colaborativa. Ele oferece uma interface amigável e recursos poderosos que permitem aos usuários criar uma ampla variedade de diagramas, incluindo organogramas, fluxos de processo, diagramas de rede, diagramas UML, entre outros. Ele é baseado em tecnologias *web* e pode ser acessado diretamente em um navegador, sem a necessidade de instalação de *software* adicional. A ferramenta é altamente flexível e compatível com várias plataformas, o que permite que os usuários acessem e trabalhem em seus diagramas a partir de qualquer dispositivo com acesso à internet.

Podemos citar como uma das principais vantagens do Draw.io é sua vasta biblioteca de elementos gráficos pré-definidos, que inclui formas, ícones e símbolos para representar diferentes elementos em um diagrama. Além disso, ele oferece recursos avançados, como alinhamento automático, linhas de conexão inteligentes, formatação de texto e estilos personalizáveis, permitindo criar diagramas visualmente atraentes e profissionais. Além disso, é possível importar e exportar diagramas em vários formatos, como PNG, PDF e XML, garantindo a portabilidade dos trabalhos realizados.

Essa ferramenta é amplamente utilizada em diversas áreas, incluindo engenharia de *software*, gerenciamento de projetos, *design* de sistemas, arquitetura empresarial e muito mais. Sua interface intuitiva e recursos avançados tornam o Draw.io uma opção popular para criar diagramas profissionais de maneira rápida e eficiente.

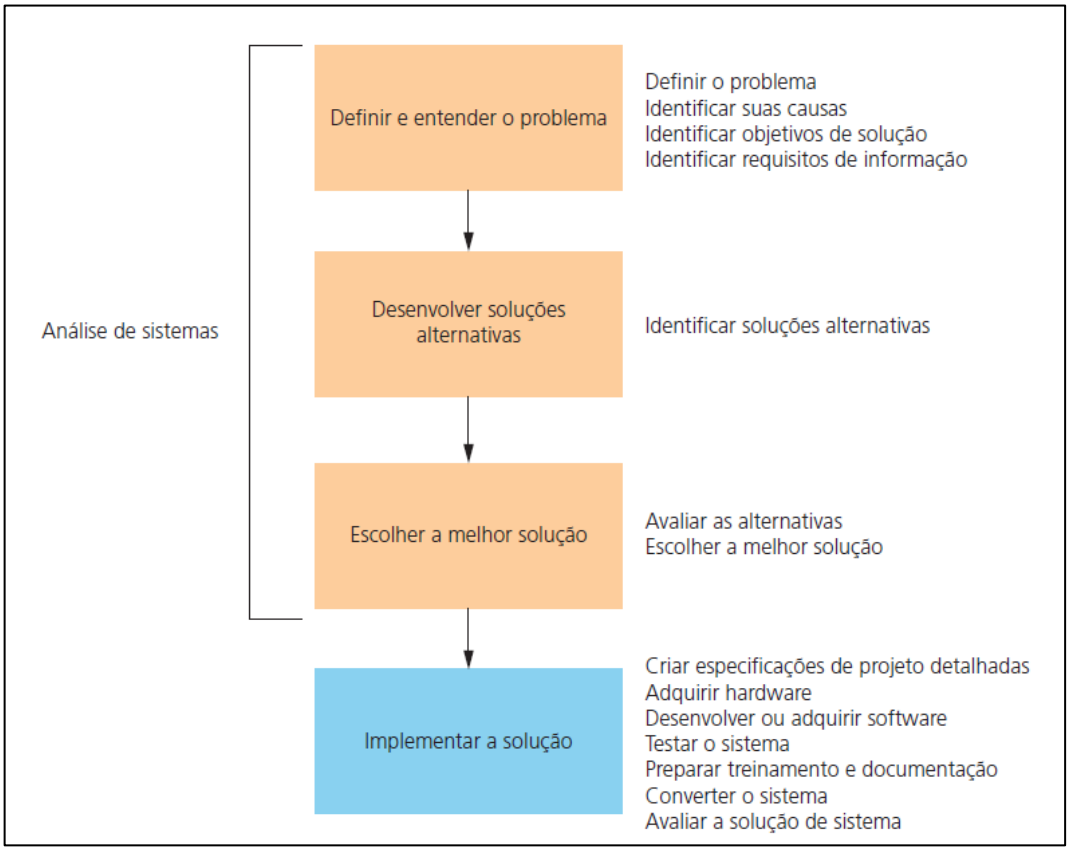
3.2 Métodos

Iremos abordar nesse tópico a resolução de problemas e desenvolvimento do sistema, conforme Laudon (2014) devemos trabalhar em quatro etapas essenciais:

1 - Definir e compreender o problema em questão;
2 - Gerar alternativas de solução;
3 - Selecionar a melhor opção;
4 - Implementar a solução escolhida.

É fundamental estabelecer uma definição adequada do problema antes de poder resolvê-lo. É necessário entender a existência do problema e sua gravidade. O problema deve ser examinado minuciosamente para que possa ser compreendido em profundidade. Em seguida, segue-se uma fase na qual são criadas diversas soluções alternativas, seguida por uma avaliação de cada uma delas para determinar a melhor opção. A etapa final consiste na implementação da solução, que envolve a elaboração de um plano detalhado para a solução, sua tradução em um sistema físico, testes, introdução na organização e, posteriormente, refinamentos conforme é utilizado (LAUDON, 2014).

Figura 13 - Desenvolvimento de uma solução de sistema de informação



Fonte: LAUDON (2014) p.404, fig.12.1

Na área de sistemas de informação, temos a análise de sistemas dividas em três primeiros passos da resolução de problemas, nos quais identificamos a questão, coletamos informações, criamos soluções alternativas e tomamos uma decisão sobre a melhor abordagem a ser adotada. Feito isso iremos implementar a solução, que no tema abordado é o desenvolvimento do *software*.

3.2.1 Metodologia eXtreme Programming - XP

Para essa atividade, com base na definição do *software*, a metodologia utilizada será a XP pois segundo Wildt et al. (2015) ela é uma metodologia ágil de desenvolvimento de *software* voltada para pequenas e médias equipes que visa melhorar a qualidade e a eficiência do processo de desenvolvimento do projeto onde os requisitos são vagos e mudam frequentemente. Ela se baseia em uma abordagem iterativa e incremental, onde as atividades são organizadas em pequenos ciclos de desenvolvimento chamados de iterações.

Essa metodologia incentiva a simplicidade no *design* e na implementação do *software*. Ela prega a utilização de práticas como refatoração contínua, onde o código é constantemente revisado e aprimorado, e a programação em pares, onde dois desenvolvedores trabalham juntos para escrever e revisar o código. Outros princípios do XP incluem a entrega frequente de *software* funcional, a melhoria contínua do processo de desenvolvimento e a adaptação às mudanças de requisitos ao longo do projeto. Tem como valores a comunicação, a simplicidade, o *feedback*, a coragem e o respeito (WILDT et al. 2015).

3.2.2 Requisitos do Sistema

Segundo Sommerville (2016) os requisitos de *software* são as especificações e descrições das funcionalidades, comportamentos e características que um sistema de *software* deve possuir para atender às necessidades de quem o utilizará (usuários e partes interessadas). Eles são fundamentais no processo de desenvolvimento de um sistema, pois servem como base para guiar todas as etapas do projeto. Existem diferentes tipos de requisitos de *software*, para esse projeto em desenvolvimento especificamente trataremos de dois deles:

1. Requisitos funcionais: São as funcionalidades e comportamentos esperados do sistema. Eles descrevem o que o *software* deve fazer e como ele deve se comportar em diferentes situações. Exemplos de requisitos funcionais incluem "o sistema deve permitir o cadastro de usuários" ou "o sistema deve calcular o total de uma compra".
2. Requisitos não funcionais: São os requisitos que definem características adicionais do sistema, como desempenho, usabilidade, segurança, confiabilidade, entre outros. Eles não se referem diretamente às funcionalidades específicas, mas são importantes para garantir a qualidade e o desempenho do *software*. Exemplos de requisitos não funcionais incluem "o sistema deve ter tempo de resposta inferior a 2 segundos" ou "o sistema deve ser compatível com navegadores X, Y e Z".

No processo de desenvolvimento de um sistema, os requisitos de *software* são coletados, analisados, documentados e validados. Eles servem como base para a

criação de documentos de especificação de requisitos e são utilizados para guiar a equipe de desenvolvimento ao longo de todo o ciclo de vida do *software*.

3.2.2.1 Requisitos Funcionais

Quadro 1 - Requisitos Funcionais

Identificação	Requisito Funcionais (RF)	Caso de Uso (UC)
RF01	Cadastro de usuários	UC01
RF02	Realizar *login* na aplicação	UC02
RF03	Cadastro e gerenciamento de tipos de materiais	UC03
RF04	Cadastro e gerenciamento de materiais	UC04
RF05	Cadastro e gerenciamento de policiais	UC05
RF06	Cadastro e gerenciamento de cautelas	UC06
RF07	Geração de relatórios	UC07
RF08	Cadastro de avisos na tela inicial	UC08

Fonte: Elaborado pelo autor (2023)

RF01: Cadastro de usuários

O sistema deverá ter um registro de utilizadores da aplicação, que deve manter um cadastro com nome completo, e-mail, senha e perfil de acesso.

RF02: Realizar *login* na aplicação

A aplicação deve ter uma validação para acesso através do cadastro dos usuários. O sistema deve exibir uma tela de *login* com e-mail e senha previamente cadastrados, exibindo a aplicação correspondente ao perfil de acesso do usuário.

RF03: Cadastro e gerenciamento de tipos de materiais

O sistema deverá manter um registro para cadastrar os tipos de materiais utilizados na OPM, por exemplo: pistola, carabina, fuzil, espingarda, colete balístico, rádio portátil (HT), etc.

RF04: Cadastro e gerenciamento de materiais

O sistema deverá manter um registro dos materiais e suas informações sendo elas: tipo de material, tombamento, número de série, marca, modelo, calibre (armas) e tamanho (coletes).

RF05: Cadastro e gerenciamento de policiais

O cadastro de policiais tomadores das cautelas deve manter as informações básicas do policial, sendo elas: RE (Registro Estatístico), nome completo e nome de guerra.

RF06: Cadastro e gerenciamento de cautelas

O sistema deve ser capaz de criar e manter o registro dos empréstimos (cautelas) utilizados pelos policiais. Esse registro deve conter: o policial utilizador do material, o material que está sendo cautelado, o dia e hora do registro da cautela, um campo opcional de observação que poderá ser editado e o registro do dia e hora da devolução do material. A aplicação também deve armazenar os dados do usuário (armeiro) que realizou o registro da cautela. Os campos policial, material, armeiro e a data da cautela não podem ser editáveis, e nenhuma cautela pode ser excluída do sistema.

RF07: Geração de relatórios

O sistema deve ser capaz de gerar relatórios das cautelas dos materiais com filtros pelos materiais, policiais e armeiros que criaram as cautelas.

RF08: Cadastro de avisos na tela inicial

O sistema deve contar em sua tela inicial uma parte destinada a exibir avisos aos armeiros referentes a atividades da função. Uma tela deverá ser disponibilizada ao usuário gestor para cadastrar e editar esses avisos.

3.2.2.2 Requisitos Não Funcionais

Quadro 2 - Requisitos Não-Funcionais

Identificação	Requisitos Não-Funcionais (RNF)
RNF01	Possuir interface simples e de fácil uso
RNF02	Possuir controle de acesso em níveis (armeiro e gestor)
RNF03	Ser compatível com quaisquer navegadores de internet
RNF04	Os relatórios devem ter a possibilidade de impressão ou salvar em PDF (*Portable Document Format*)

Fonte: Elaborado pelo autor (2023)

RNF01: Possuir interface simples e de fácil uso

O sistema deve adotar uma interface de utilização simples, sem muitos botões ou informações que possam confundir o usuário, somente o necessário para utilização deve ser implementado

RNF02: Possuir controle de acesso em níveis (armeiro e gestor)

A aplicação deve possuir um perfil de acesso: Gestor ou Armeiro (administrador ou usuário). O gestor deverá ter acesso total de administração do sistema. Já o armeiro terá um perfil de usuário, onde apenas irá inserir informações de cadastro de materiais e de cautelas.

RNF03: Ser compatível com quaisquer navegadores de internet

A aplicação deve ser desenvolvida para ambiente *desktop*, compatível com os navegadores mais utilizados e sua estrutura deve ser adaptativa a resolução da tela que o usuário estiver fazendo uso.

RNF04: Os relatórios devem ter a possibilidade de impressão ou salvar em PDF

Os relatórios emitidos pelo usuário: gestor, devem possuir a possibilidade de ser impresso e salvos em formato PDF - formato de documento portátil.

3.2.3 Modelagem do Sistema

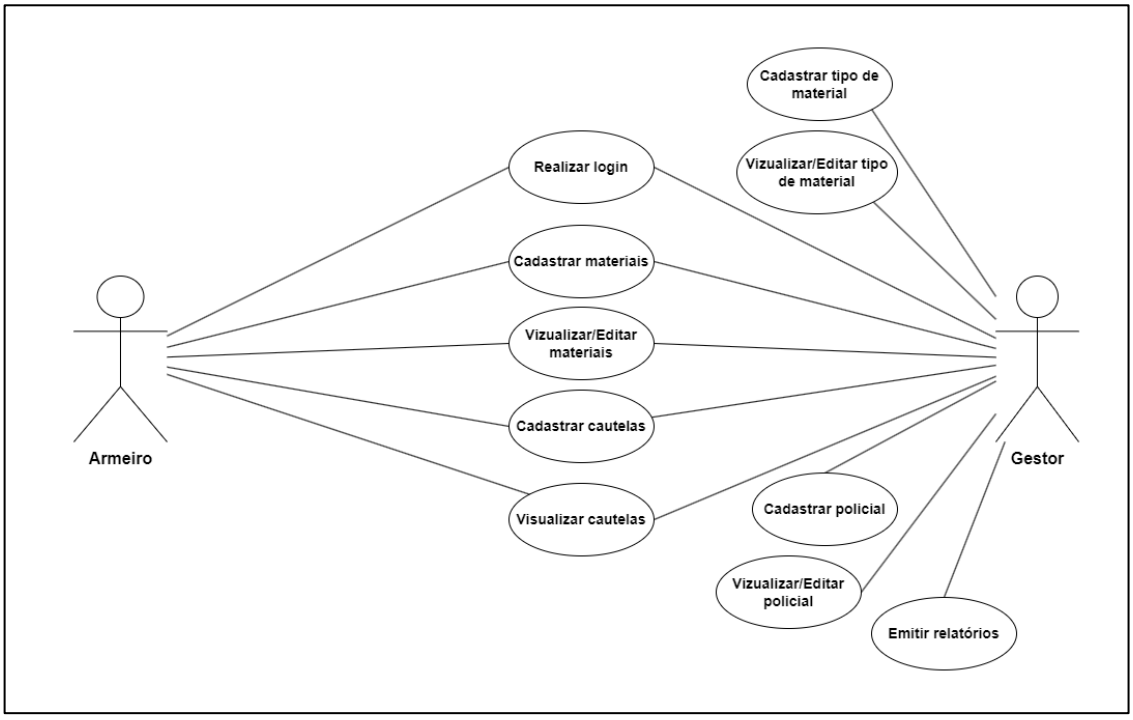
Após o levantamento dos requisitos do *software* trataremos da modelagem do sistema. Ian Sommerville (2016) aborda a modelagem do sistema como um processo que envolve a criação de representações abstratas do sistema em desenvolvimento. A modelagem é utilizada para entender, analisar e comunicar as características e comportamentos do sistema, permitindo que os desenvolvedores e as partes interessadas tenham uma visão mais clara do sistema antes de sua implementação. A modelagem do sistema inclui a criação de diferentes tipos de modelos, como modelos de requisitos, modelos de arquitetura, modelos de dados, modelos de processos, entre outros. Cada modelo desempenha um papel específico na compreensão do sistema e na facilitação de diferentes atividades de desenvolvimento.

A modelagem do sistema também pode servir como base para a geração de código ou a criação de protótipos. A partir dos modelos, é possível automatizar a geração de parte do código-fonte ou desenvolver protótipos funcionais que ajudam a validar os requisitos e as decisões de *design*. Em suma, a modelagem do sistema, de acordo com Sommerville (2016), é o processo de criação de representações abstratas do sistema que auxiliam na compreensão, análise, comunicação e validação do sistema em desenvolvimento.

3.2.3.1 Diagrama de Caso de Uso

A modelagem do sistema por meio dos casos de uso é uma técnica utilizada no desenvolvimento de *software* para identificar, descrever e visualizar as interações entre os atores (usuários, sistemas externos, dispositivos, etc.) e o sistema em questão. Os casos de uso são representações de cenários que descrevem as funcionalidades e os comportamentos do sistema a partir da perspectiva do usuário. Um caso de uso é composto por atores, que são entidades externas que interagem com o sistema, e por uma sequência de passos que descrevem as ações e as respostas esperadas durante a interação entre o ator e o sistema. Esses casos de uso ajudam a capturar os requisitos funcionais do sistema de forma clara e compreensível.

Figura 14 - Diagrama de caso de uso



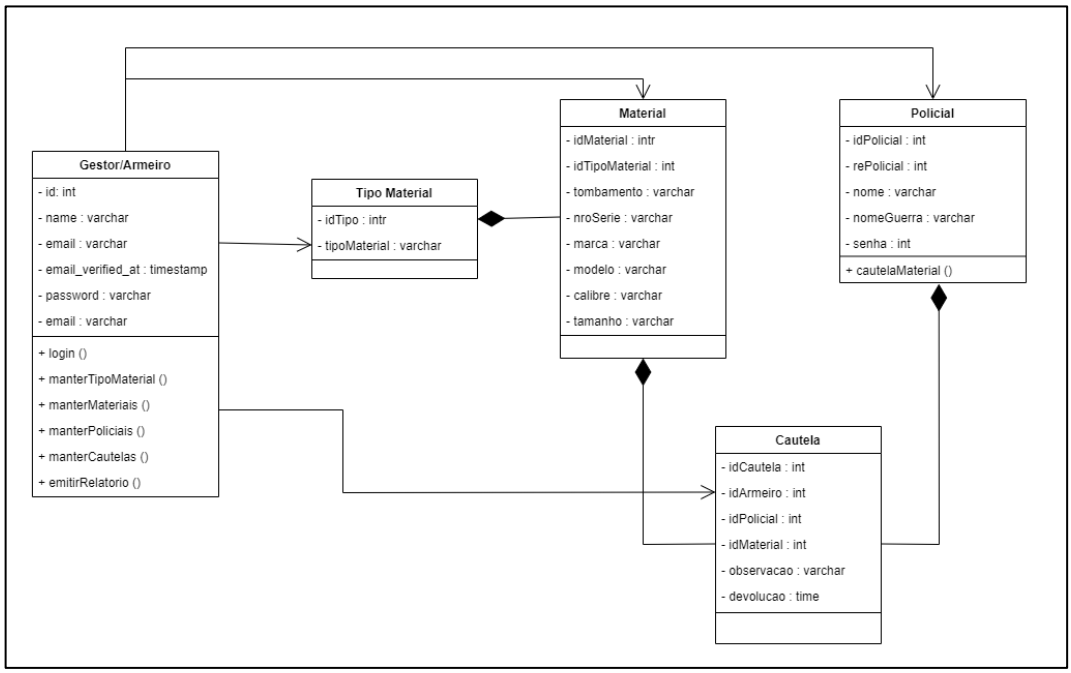
Fonte: Elaborado pelo autor (2023)

3.2.3.2 Diagrama de Classe

A modelagem do sistema por meio do diagrama de classe é a técnica utilizada no desenvolvimento para representar a estrutura e as relações entre as classes de um sistema. O diagrama de classe é uma das ferramentas da UML que permite visualizar as entidades do sistema, seus atributos e os relacionamentos entre elas. No diagrama de classe, as classes são representadas por retângulos, com os atributos e métodos listados dentro do retângulo. As relações entre as classes são mostradas por meio de linhas que conectam os retângulos, indicando associações, heranças, agregações e outras formas de relacionamento.

A modelagem do sistema por meio do diagrama de classe tem como objetivo principal representar a estrutura estática do sistema, ou seja, as classes, seus atributos e métodos, e como elas se relacionam entre si. Isso proporciona uma visão clara da organização do sistema e das interações entre suas partes.

Figura 15 - Diagrama de classes



Fonte: Elaborado pelo autor (2023)

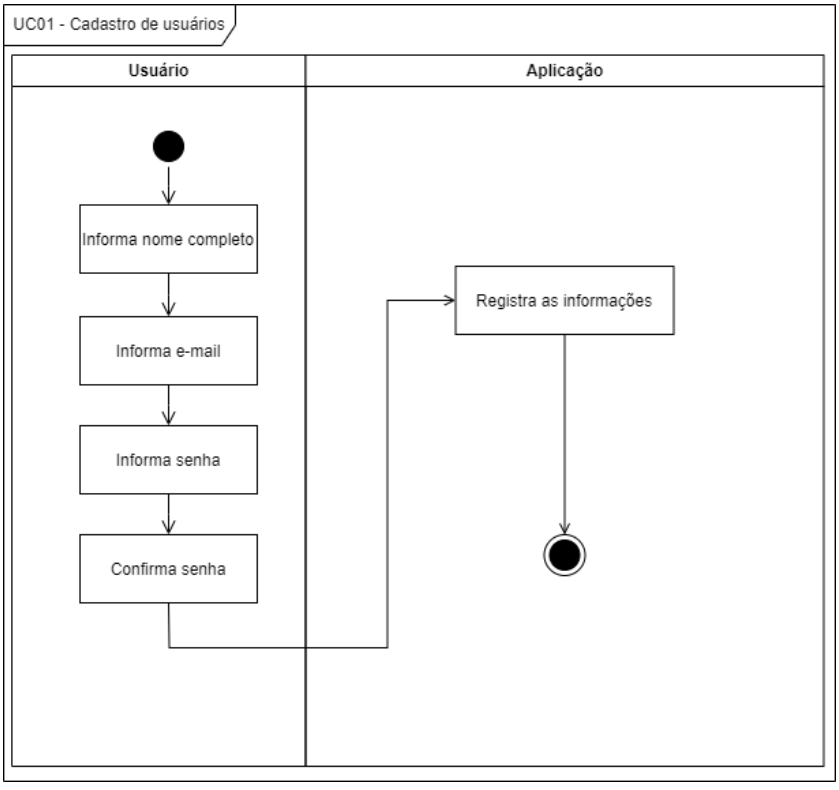
3.2.3.3 Diagramas de Atividade

A modelagem do sistema por meio do diagrama de atividades é a técnica utilizada no desenvolvimento para representar o fluxo de atividades, a sequência de ações e as decisões tomadas em um processo. O diagrama de atividades é uma das

ferramentas da UML que permite visualizar as atividades, as transições e as ramificações em um sistema. As atividades são representadas por retângulos, com as ações e as decisões listadas dentro do retângulo. As transições entre as atividades são mostradas por meio de setas que indicam a sequência de execução. Além disso, o diagrama de atividades também permite representar o fluxo de controle, as ramificações condicionais e as atividades paralelas. Isso envolve identificar as atividades envolvidas, definir as ações realizadas em cada atividade e especificar as decisões tomadas com base em condições específicas. Ele também auxilia na compreensão e na comunicação entre os membros da equipe de desenvolvimento, fornecendo uma visão clara do fluxo de execução e das interações entre as atividades.

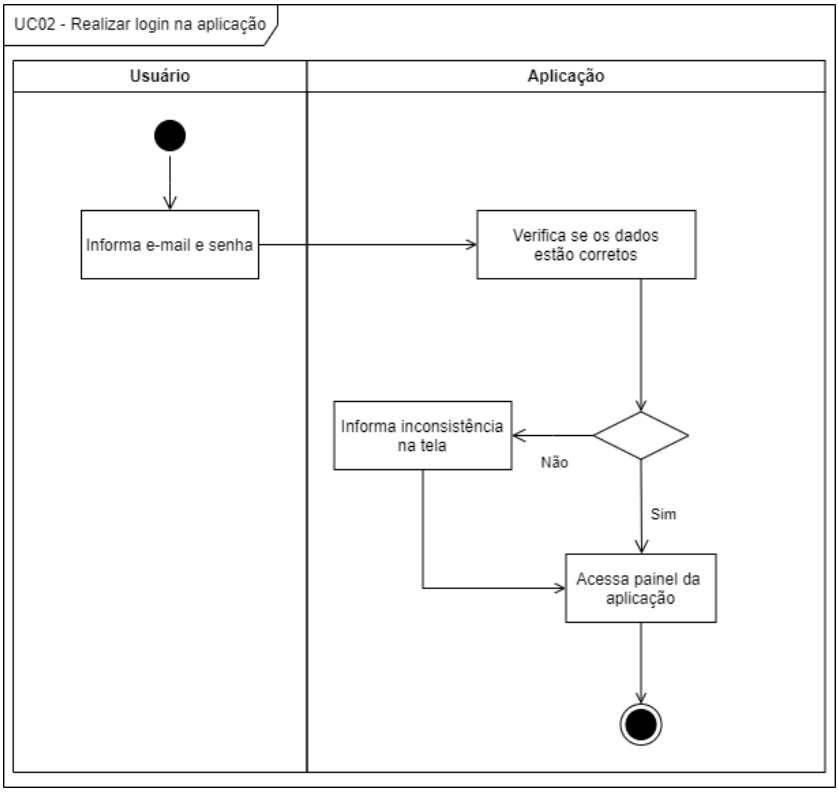
Abaixo são demonstrados os diagramas de atividade baseados nos casos de uso apresentados no Quadro 1 – Requisitos funcionais.

Figura 16 - Diagrama de atividade UC01



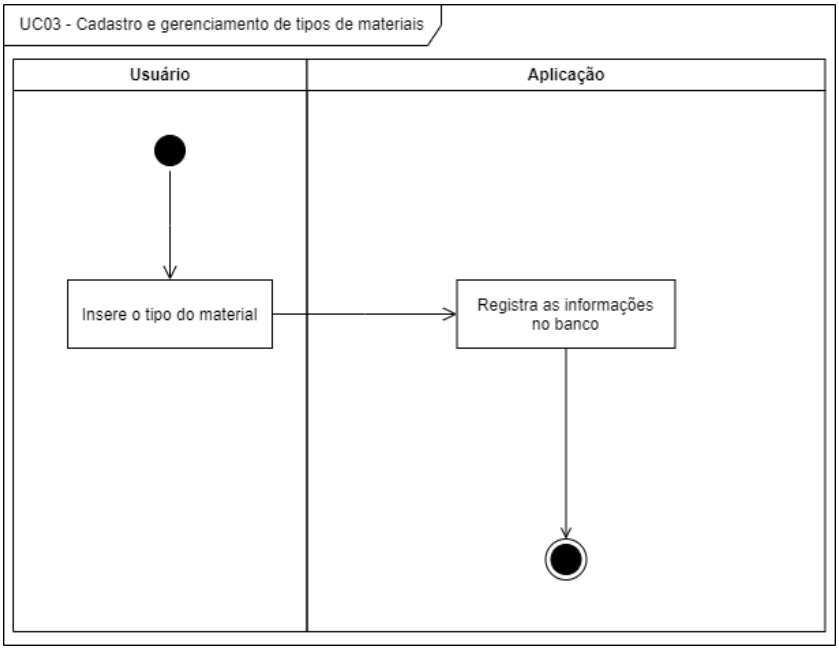
Fonte: Elaborado pelo autor (2023)

Figura 17 - Diagrama de atividade UC02



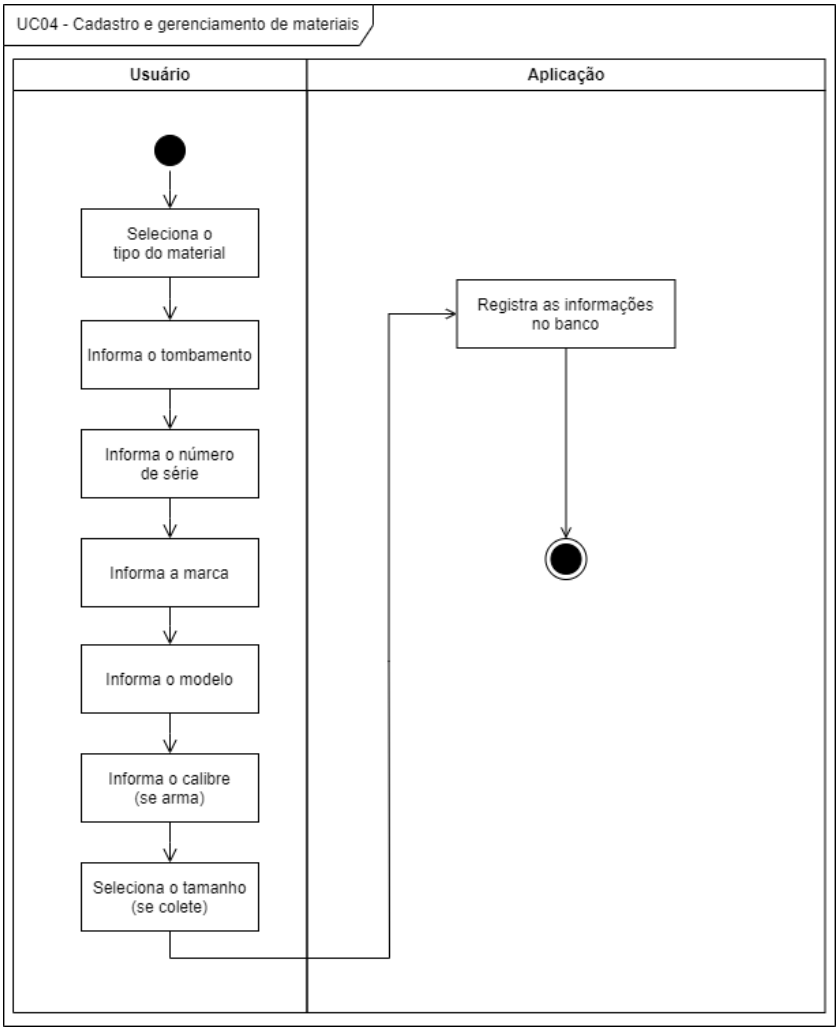
Fonte: Elaborado pelo autor (2023)

Figura 18 - Diagrama de atividade UC03



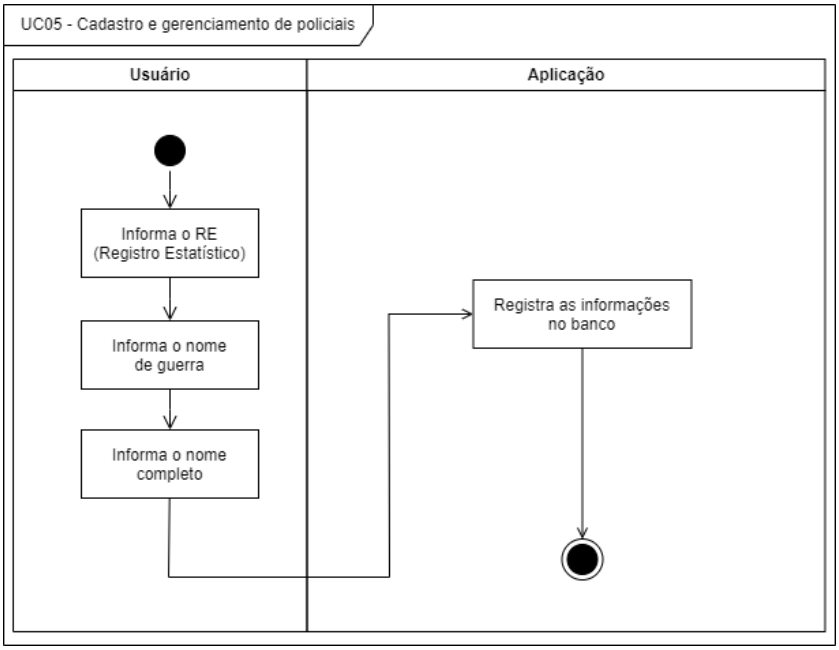
Fonte: Elaborado pelo autor (2023)

Figura 19 - Diagrama de atividade UC04



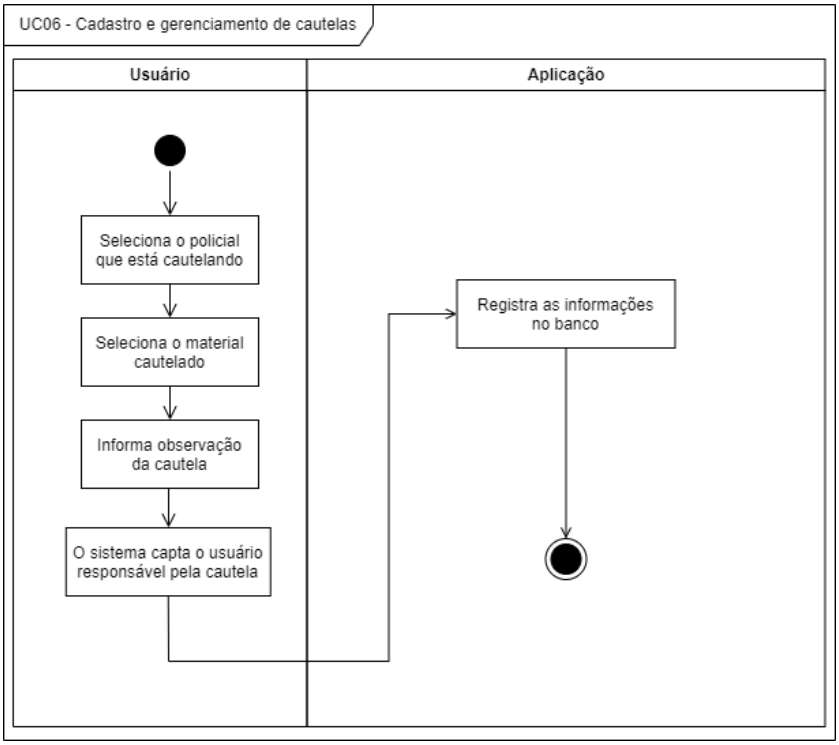
Fonte: Elaborado pelo autor (2023)

Figura 20 - Diagrama de atividade UC05



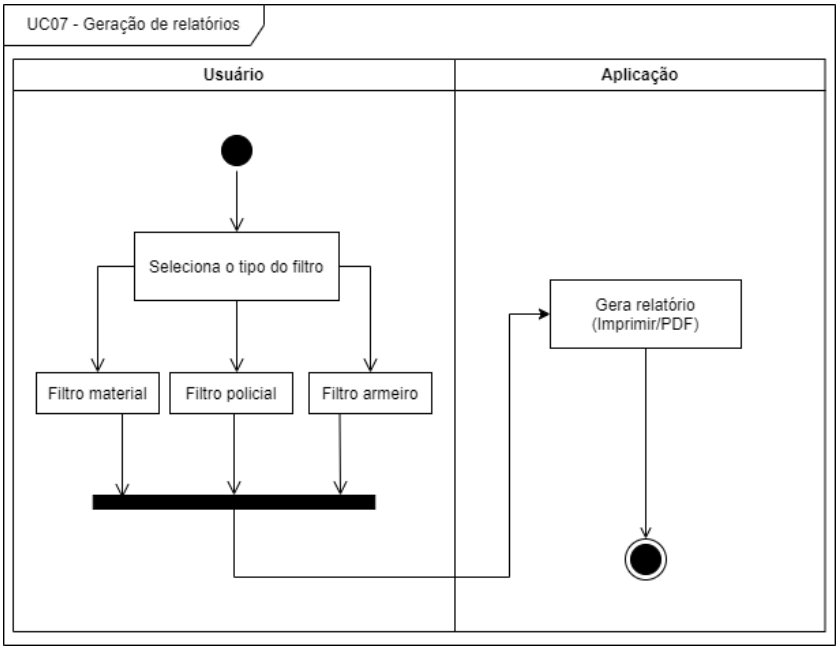
Fonte: Elaborado pelo autor (2023)

Figura 21 - Diagrama de atividade UC06



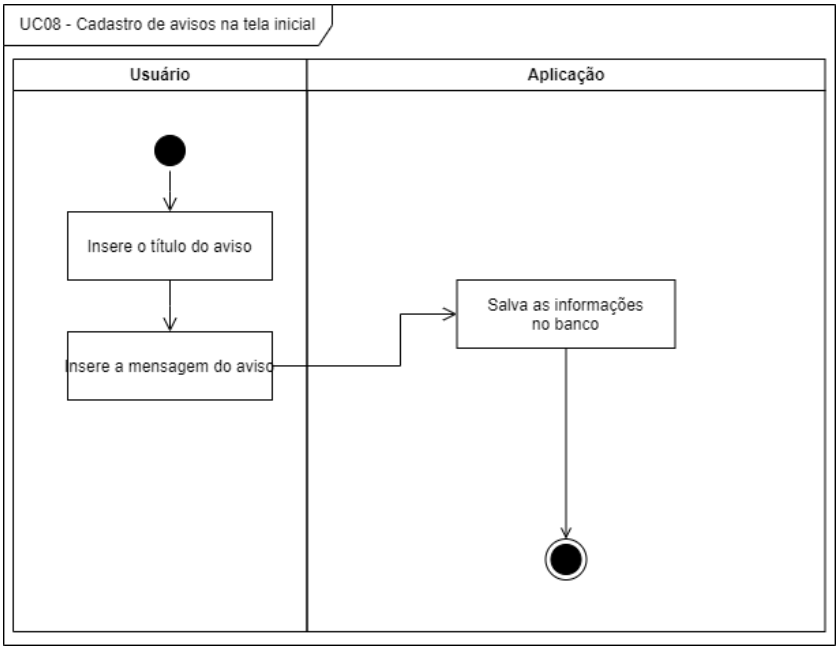
Fonte: Elaborado pelo autor (2023)

Figura 22 - Diagrama de atividade UC07



Fonte: Elaborado pelo autor (2023)

Figura 23 - Diagrama de atividade UC08



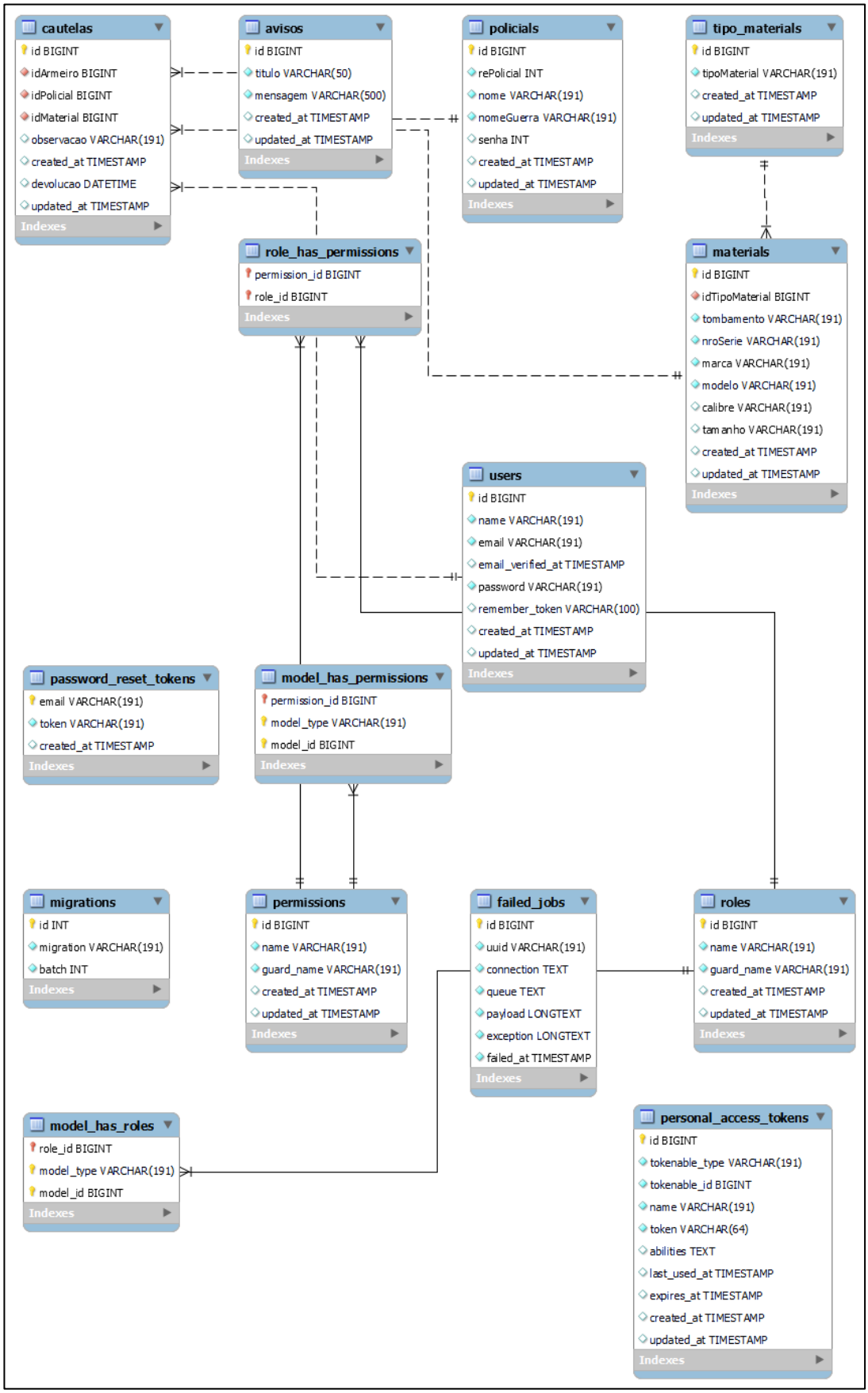
Fonte: Elaborado pelo autor (2023)

3.2.3.4 Diagrama de Entidade Relacionamento

A modelagem do sistema por meio do diagrama Entidade Relacionamento - ER é uma técnica utilizada no desenvolvimento para representar a estrutura e as relações entre as entidades e os atributos de um banco de dados. O diagrama ER é uma ferramenta gráfica que permite visualizar as entidades do sistema, os atributos que descrevem essas entidades e as associações entre elas. As entidades são representadas por retângulos, com os atributos listados dentro do retângulo. As associações entre as entidades são mostradas por meio de linhas que conectam os retângulos, indicando os relacionamentos, como associações, cardinalidades e restrições.

A modelagem do sistema por meio do diagrama ER tem como objetivo principal representar a estrutura e as relações dos dados no sistema. Isso envolve identificar as entidades relevantes, definir seus atributos e especificar as associações entre elas. Essa representação visual facilita a compreensão da estrutura do banco de dados e ajuda no planejamento e no projeto do sistema.

Figura 24 - Diagrama de entidade relacionamento



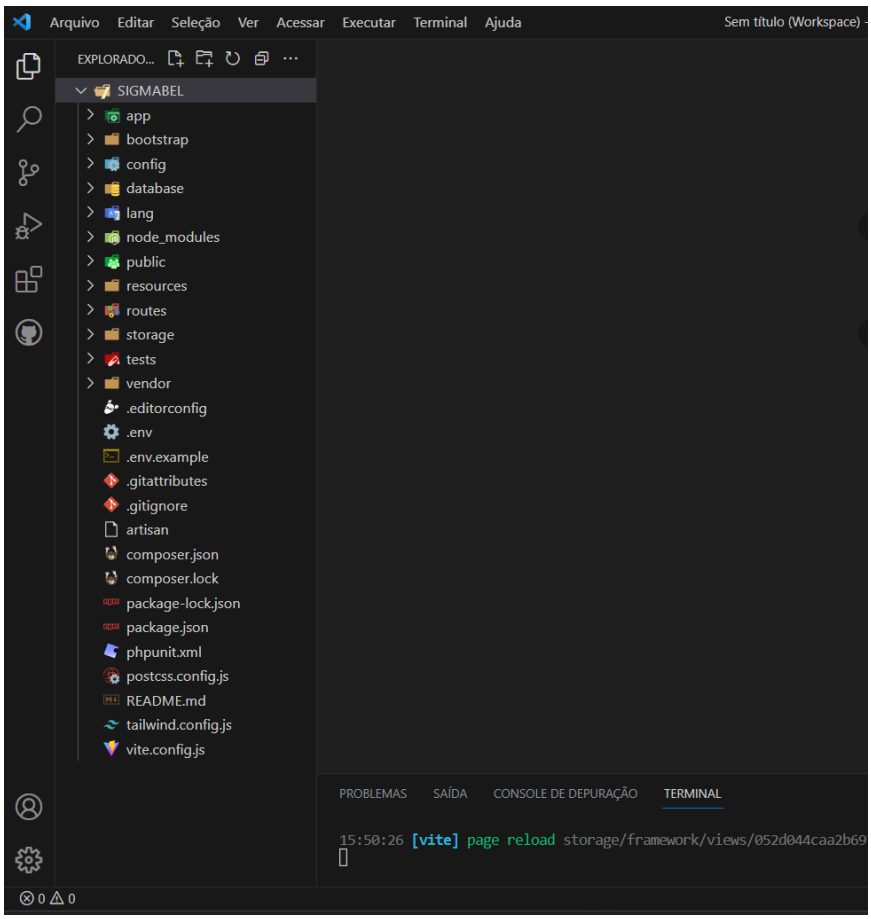
Fonte: Elaborado pelo autor (2023)

4 RESULTADOS

Neste capítulo é apresentado o processo de desenvolvimento da aplicação *web* descrita neste trabalho e também a sua concepção, apresentando telas e comentários do processo de elaboração.

A construção do SIGMABEL iniciou-se com a instalação dos materiais, o primeiro, o Laragon versão 6.0 é o *software* responsável pela maior parte das ferramentas, pois disponibiliza todo o ambiente de desenvolvimento, sendo instalado através dele o Laravel 10, o PHP versão 8.1.10, o Apache versão 2.4.54, o MySQL versão 8.0.30 e o HeidSQL versão 12.1 *portable*. Com o projeto Laravel criado através do *composer e* estruturado (Figura 25), utilizamos o VSCode para iniciar a configuração da aplicação, criação e também a edição do *software*, na forma de código, na linguagem de programação PHP e a linguagem de marcação HTML.

Figura 25 - Estrutura de pastas do projeto



Fonte: Elaborado pelo autor (2023)

Após a configurado do ambiente, iniciamos com a criação das páginas no diretório do projeto em linguagem PHP, durante todo o processo de desenvolvimento foram realizados inúmeros testes com o Google Chrome para garantir o perfeito funcionamento da aplicação *web*, o Laragon é capaz de criar um endereço local com nome do projeto com a terminação “.test”, facilitando o acesso, a programação e os testes devido as requisições HTTP que se comunicam através das rotas URL.

É fundamental apresentar os resultados obtidos após a implementação e testes da aplicação descrita no projeto. Essa etapa visa demonstrar de forma clara e objetiva o funcionamento da solução desenvolvida, bem como a validação dos objetivos propostos.

Figura 26 - Tela inicial da aplicação

Fonte: Elaborado pelo autor (2023)

4.1 Metodologia e Padrões Escolhidos

O uso do SIG aplicado a esse *software web* criado através do Laravel, nos padrões MVC, trouxe diversos benefícios e pontos positivos para a gestão de uma reserva de armamentos de uma OPM. A seguir, apresento seis pontos positivos desse contexto:

- ✓ Organização e estrutura: O padrão MVC proporcionou uma separação clara das responsabilidades do *software*, dividindo-o em camadas de modelo, visão e controle. Isso resultou em um código mais organizado, facilitando a manutenção, a reutilização e a escalabilidade do sistema.
- ✓ Agilidade no desenvolvimento: O Laravel é um *framework* PHP conhecido por sua produtividade e facilidade de desenvolvimento. Ao utilizar esse *framework*, foi possível acelerar o processo de criação do *software*, aproveitando recursos prontos e bibliotecas disponíveis.
- ✓ Interface intuitiva e amigável: A aplicação do padrão MVC permite separar a lógica de negócio da camada de apresentação (*view*). Dessa forma, é possível criar interfaces intuitivas, com *layouts* atraentes e interações amigáveis, proporcionando uma melhor experiência de uso para os usuários do sistema.
- ✓ Flexibilidade e extensibilidade: O Laravel proporciona uma arquitetura modular, que facilita a adição de novas funcionalidades ao sistema. É possível desenvolver novos módulos de forma independente e integrá-los ao *software* principal de maneira eficiente, tornando o sistema mais flexível e adaptável às necessidades da organização.
- ✓ Gerenciamento eficiente de dados: O uso de um SIG aplicado em um *software web* criado com Laravel possibilita o gerenciamento eficiente de dados. Com o *model* de dados bem estruturado (modelo), é possível armazenar, acessar e manipular informações relevantes para a tomada de decisões gerenciais.
- ✓ Relatórios e análises: Um dos principais objetivos da escolha do SIG é fornecer informações relevantes para a gestão. Com esse sistema, é possível desenvolver recursos de geração de relatórios e análises que facilitam o acompanhamento dos indicadores do uso, o monitoramento dos materiais bélicos e a tomada de decisões estratégicas.

Esses são apenas alguns dos pontos positivos do uso de Sistemas de Informações Gerenciais aplicados em um *software web* criado com Laravel. A combinação dessas tecnologias proporciona uma base sólida para o desenvolvimento de soluções gerenciais eficientes, permitindo uma gestão mais eficaz e embasada em informações precisas.

4.2 Sistema Desenvolvido

O sistema *web* desenvolvido denominado SIGMABEL, leva em consideração todos os requisitos definidos anteriormente. Foram utilizados os padrões propostos, a aplicação das cores da PM-RO, a exibição do brasão da instituição e a logo do projeto. A seguir, veremos as funcionalidades que foram desenvolvidas no sistema, demonstrando as telas e as interações disponíveis para os usuários, bem como as ações executadas e os resultados obtidos.

Abaixo listamos todas as telas desenvolvidas para aplicação:

Tela inicial

- ✓ Logo da aplicação
- ✓ Botões de login e cadastro

Tela de login

- ✓ Plano de fundo personalizado
- ✓ Campos de e-mail e senha para acesso

Tela de cadastro de usuário

- ✓ Plano de fundo personalizado
- ✓ Campos para inserção: nome completo, e-mail, senha e confirmação da senha

Tela de Inicial com informações

- ✓ Tela principal do sistema
- ✓ Gráfico de barras com todos materiais cadastrados e quantidade
- ✓ Mural de avisos, onde o gestor pode publicar informações para o usuário

Tela dos Tipos de Materiais

- ✓ Exibe todos os tipos de materiais cadastrados no sistema
- ✓ Possui botão de visualizar individualmente cada um para editar

Tela de edição de Tipo de Material

- ✓ Campo para editar o tipo de material
- ✓ Botões de salvar, voltar e excluir

Tela de cadastro de Tipo de Material

- ✓ Campos para inserir o tipo do material a ser cadastrado
- ✓ Botões salvar e voltar página

Tela dos Materiais

- ✓ Exibe todos materiais cadastrados no sistema
- ✓ Possui botão de visualizar individualmente cada um para editar

Tela de edição dos Material

- ✓ Campos para editar os dados do material
- ✓ Botões de salvar, voltar e excluir

Telas de cadastro de Material

- ✓ Campos para inserir dados material a ser cadastrado
- ✓ Botões salvar e voltar página

Tela dos Policiais

- ✓ Exibe todos policiais cadastrados no sistema
- ✓ Possui botão de visualizar individualmente cada um para editar

Tela de edição dos Policial

- ✓ Campos para editar os dados do policial
- ✓ Botões de salvar, voltar e excluir

Tela de cadastro de Policial

- ✓ Campos para inserir os dados do policial a ser cadastrado
- ✓ Botões salvar e voltar página

Tela de Cautelas

- ✓ Exibe todas cautelas cadastradas no sistema
- ✓ Possui botão de visualizar individualmente cada uma para editar

Tela de edição das Cautela

- ✓ Campos para cadastrar devolução e observações da cautela
- ✓ Botões de salvar e voltar

Tela de cadastro de Cautela

- ✓ Campos para selecionar o policial e o material da cautela
- ✓ Botões salvar e voltar página

Tela de emissão de Relatórios

- ✓ Três tipos de filtros das cautelas (material, policial e armeiro)
- ✓ Respectivos botões para acessar

Tela de impressão dos Relatórios

- ✓ Exibe os dados filtrados da cautela
- ✓ Ícone de impressão/salvar relatório

Tela dos Avisos

- ✓ Exibe todos avisos exibidos na tela inicial do sistema
- ✓ Possui botão de visualizar individualmente cada uma para editar

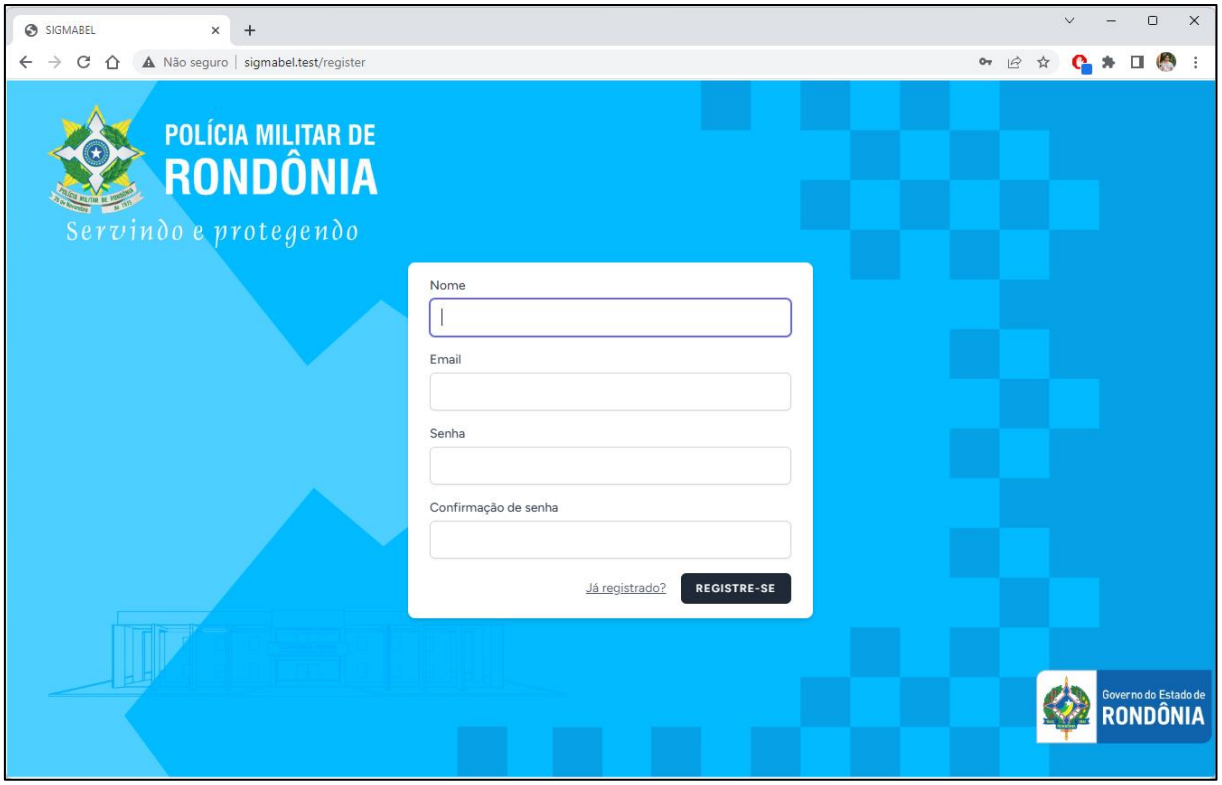
Tela de edição do Aviso

- ✓ Campos para editar os dados do aviso
- ✓ Botões de salvar, voltar e excluir

Tela de cadastro de Avisos

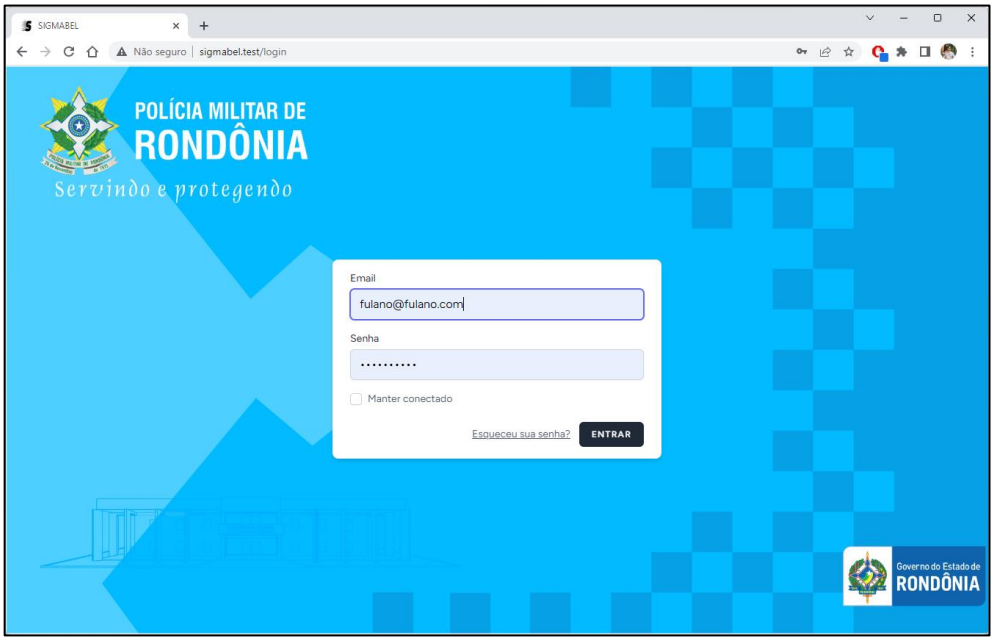
- ✓ Campos para inserir título e mensagem
- ✓ Botões salvar e voltar página

Figura 27 - Tela de cadastro do usuário



Fonte: Elaborado pelo autor (2023)

Figura 28 - Tela de login



Fonte: Elaborado pelo autor (2023)

Figura 29 - Tela de informações e avisos

Fonte: Elaborado pelo autor (2023)

Figura 30 - Tela dos tipos de materiais cadastrados

Fonte: Elaborado pelo autor (2023)

Figura 31 - Tela para cadastro do tipo de material

Fonte: Elaborado pelo autor (2023)

Figura 32 - Tela dos materiais cadastrados

Fonte: Elaborado pelo autor (2023)

Figura 33 - Tela para cadastro de material

Fonte: Elaborado pelo autor (2023)

Figura 34 - Tela para edição de material

Fonte: Elaborado pelo autor (2023)

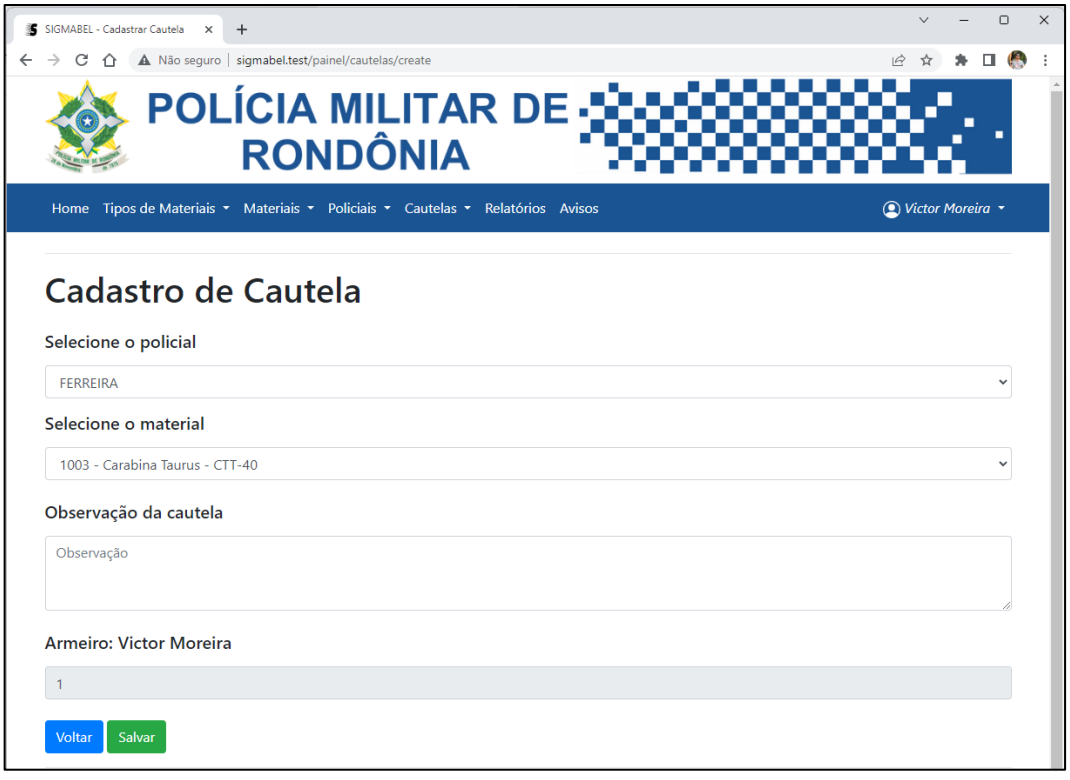
Figura 35 - Tela dos policiais cadastrados

Fonte: Elaborado pelo autor (2023)

Figura 36 - Tela das cautelas cadastradas

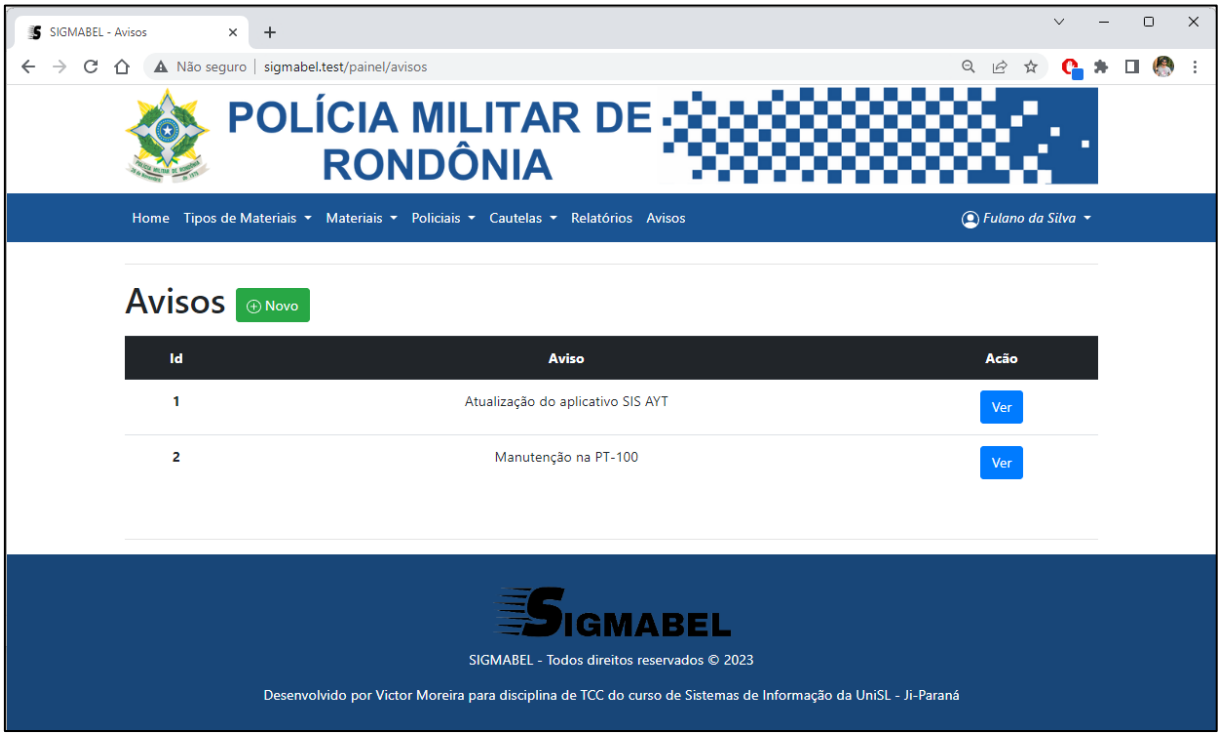
Fonte: Elaborado pelo autor (2023)

Figura 37 - Tela para cadastro de cautela



Fonte: Elaborado pelo autor (2023)

Figura 38 - Tela de avisos



Fonte: Elaborado pelo autor (2023)

Figura 39 - Tela cadastro de aviso

Fonte: Elaborado pelo autor (2023)

4.3 Relatórios Gerados Pelo Sistema

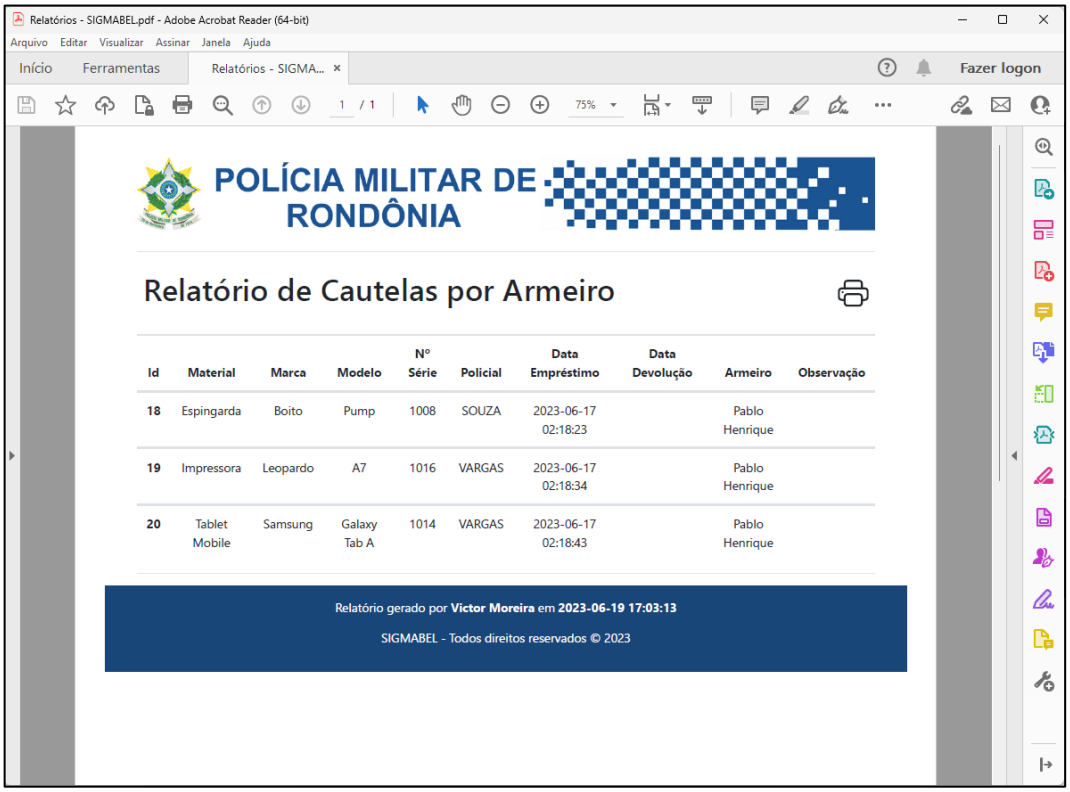
A escolha do SIG para elaboração desse sistema se deu devido a esse padrão de Sistema de Informação possibilitar a emissão de relatórios com informações relevantes para a gestão. A geração de relatórios, com filtros, da base de dados possibilita a pessoa responsável pelos materiais ter um controle melhor do patrimônio e realizar análises que facilitam a tomada de decisões e o rastreamento de uso do material.

Abaixo na Figura 37, podemos ver um relatório gerado e salvo no formato PDF com um filtro aplicado que retorna da base de dados os registros de todas as cautelas que determinado armeiro foi responsável por realizar o empréstimo.

Figura 40 - Tela de emissão de relatórios

Fonte: Elaborado pelo autor (2023)

Figura 41 – Relatório por armeiro emitido em PDF



Fonte: Elaborado pelo autor (2023)

Figura 42 - Relatório por material emitido em PDF

Fonte: Elaborado pelo autor (2023)

5 CONCLUSÃO

Percebemos nos dias atuais que as tecnologias digitais estão se inserindo cada vez mais na gestão pública, pois fazem parte de um planejamento estratégico que vem acontecendo ao decorrer do tempo. A aplicação da tecnologia da informação nas organizações públicas deve acontecer assim como nas organizações privadas visando dar qualidade a infraestrutura e aos serviços públicos. Na gestão privada visa-se o lucro e na gestão pública a proposta é entregar melhor prestação de serviço a toda população. Vale-se destacar que a Constituição Federal em seu art. 218 descreve que o Estado promoverá e incentivará o desenvolvimento científico, a pesquisa, a capacitação científica e tecnológica e a inovação, com isso a participação dos cidadãos nesse desenvolvimento é de suma importância realizando um planejamento participativo.

A adoção do Laravel 10, um framework PHP robusto e moderno, em conjunto com a metodologia XP, trouxe agilidade ao processo de desenvolvimento, possibilitando a entrega contínua de funcionalidades e a adaptação ágil às necessidades e requisitos da aplicação. Isso resultou em um sistema eficiente tanto para os usuários quanto para a organização envolvida. O framework ofereceu recursos avançados para a modelagem de bancos de dados, implementação de autenticação e autorização, além da geração de relatórios, o que facilitou a criação de um sistema completo e funcional. É importante ressaltar que a substituição de processos manuais realizados por seres humanos por sistemas computadorizados traz uma série de vantagens e benefícios significativos. Os sistemas computadorizados são capazes de executar tarefas de maneira rápida e precisa, o que resulta em maior eficiência e produtividade. Enquanto a intervenção humana está sujeita a erros e falhas, os sistemas computadorizados realizam tarefas de forma consistente e precisa. Isso é especialmente relevante em processos que envolvem gerenciamento de dados ou outras operações críticas. Além disso, o novo sistema permite o armazenamento centralizado e seguro dos dados, garantindo um acesso rápido e fácil às informações necessárias. Isso suporta análises e tomadas de decisão embasadas em informações atualizadas e confiáveis. A centralização dos dados também contribui para a integridade e segurança das informações, evitando perdas ou manipulações indevidas.

Por fim, vale ressaltar a oportunidade desse projeto para o bacharelando em Sistemas de Informação, que também exerce a profissão de Policial Militar. Essa integração entre conhecimento acadêmico e prática profissional permite a aplicação direta do aprendizado do curso, além de cumprir seu papel de cidadão, tendo a oportunidade de aplicar padrões de projetos, de governo eletrônico e novos meios e formas de se desenvolver um sistema de gestão pública.

Destaca-se ainda que esse projeto envolve mudanças culturais, e não pode ser considerado um projeto acabado, pois estão sempre em franco desenvolvimento participativo. E devido ele ser escalável e flexível, isto é, ele pode ser facilmente adaptado e personalizado de acordo com as necessidades específicas do local empregado, permitindo a incorporação de novas funcionalidades e acomodando mudanças nos processos.

REFERÊNCIAS

BENTO, E. J. **Desenvolvimento web com PHP e MySQL**. São Paulo: Casa do Código, 2018. 204 p.

BOOTSTRAP. Disponível em:<https://getbootstrap.com.br/docs/4.1/about/overview>. Acesso em 24 de mar 2023.

BRASIL. [Constituição (1988)]. **Constituição da República Federativa do Brasil de 1988**. Brasília: Senado Federal, Coordenação de Edições Técnicas, 2016. Disponível em: <https://www2.senado.leg.br/bdsf/bitstream/handle/id/518231/CF88_Livro_EC91_2016.pdf>. Acesso em: 22 abr. 2022.

BRASIL. Decreto nº 9.847, de 25 de junho de 2019. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/decreto/D9847.htm>. Acesso em: 19 fev. 2022.

BRASIL. Decreto nº 10.030, de 30 de setembro de 2019. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/decreto/D10030.htm>. Acesso em: 19 fev. 2022.

BRASIL. Decreto nº 79.108, de 11 de janeiro de 1977. Disponível em: < https://www.planalto.gov.br/ccivil_03/decreto/1970-1979/d79108.htm>. Acesso em: 19 fev. 2022.

BRASIL. Lei nº 6.270, de 24 de novembro de 1975. Disponível em: < https://www.planalto.gov.br/ccivil_03/leis/1970-1979/l6270.htm>. Acesso em: 19 fev. 2022.

BRASIL. Lei nº 10.826, de 22 de dezembro de 2003. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2003/l10.826.htm>. Acesso em: 19 fev. 2022.

CARVALHO, V. **MySQL: Comece com o Principal Banco de Dados Open Source do Mercado**. São Paulo: Casa do Código, 2015. 214 p.

EXÉRCITO BRASILEIRO. Portaria n° 07 D-LOG, de 28 de abril de 2006. Disponível em: <http://www.dfpc.eb.mil.br/images/Portaria007DLog_28Abr06.pdf>. Acesso em: 19 fev. 2022.

FOWLER, M. et al. **Padrões de arquitetura de aplicações corporativas**. Porto Alegre: Bookman, 2007. Tradução: Acauan Fernandes. 489 p.

GOUVEIA, L. B. **Gestão da informação para transformação digital**. Belo Horizonte: Conhecimento Editora, 2023. 246 p.

GUEDES, G. T. A. **UML 2: Uma Abordagem Prática**. 2. ed. São Paulo: Novatec Editora, 2011. 488 p.

HEIDISQL. Disponível em: <https://www.heidisql.com/>. Acesso em: 22 abr. 2022.

LAUDON, K. C. LAUDON, J. P. **Sistemas de Informação Gerenciais**. 11. ed. São Paulo: Pearson Education do Brasil, 2014. Tradução: Célia Taniwaki. 484 p.

LARAGON. Disponível em: <https://laragon.org/docs/>. Acesso em: 22 abr. 2022.

MARIANO, D. **Bootstrap 5: Guia rápido para iniciantes.** 1. ed. Lagoa Santa: Alfahelix, 2022. 108 p.

MINISTÉRIO DA JUSTIÇA E DA SEGURANÇA PÚBLICA. **Identificação de armas de fogo e munições**. 1. ed. Brasília: Secretaria Nacional de Segurança Pública, 2018.

MUSEU VIRTUAL DA PMRO. Disponível em: <https://museu.pm.ro.gov.br>. Acesso em: 18/05/2023.

OLIVEIRA, D. P. R. **Sistemas de informações gerenciais:** estratégias, táticas, operacionais. 17. ed. São Paulo: Atlas, 2018. 314 p.

PILGRIM, M. **HTML5: Entendendo e executando**. 1. ed. Rio de Janeiro: Alta Books, 2012. 220 p.

PIRES, J. B. et al. **Controle de Material Bélico no Âmbito da PMRO**. Porto Velho: 2020, 28 p. Disponível: <http://dspace.mj.gov.br/handle/1/4658>. Acesso em: 19 fev. 2022.

PRESSMAN, R. S. MAXIM, B. R. **Engenharia de Software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016. Tradução: João Eduardo Nóbrega Tortello; Revisão técnica: Reginaldo Arakaki, Júlio Arakaki, Renato Manzan de Andrade. 940 p.

REZENDE, D. A. ABREU, A. F. **Tecnologia da Informação Aplicada a Sistemas de Informação Empresariais:** o papel estratégico da informação e dos sistemas de informação nas empresas. 9. ed. São Paulo: Atlas, 2013. 345 p.

SILBERSCHATZ, A. KORTH, H. F. SUDARSHAN, S. **Sistema de Banco de Dados**. 6. ed. São Paulo: Editora LTC, 2011. 904 p.

SOMMERVILLE, I. **Engenharia de Software**, 9. ed. São Paulo: Pearson, 2011. Tradução: Ivan Bosnic e Kalinka G. de O. Gonçalves. Revisão técnica Kechi Hirama. 529 p.

THE PHP GROUP. Disponível em: <https://www.php.net/manual/pt_BR/intro-whatis.php>. Acesso em: 22 abr. 2022.

VISUAL STUDIO CODE. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 22 abr. 2022.

WILDT. D. et al. **Extreme programming: Práticas para o dia a dia no desenvolvimento ágil**. São Paulo: Casa do Código, 2015. 144 p.

W3SCHOOLS, HTML. Disponível em: <https://www.w3schools.com/html/html_intro.asp>. Acesso em: 22 abr. 2022.

APÊNDICE A – DICIONÁRIO DE DADOS

Neste apêndice contém o dicionário de dados com base no diagrama de entidade relacionamento apresentado na Figura 24. Nos quadros abaixo são descritas as seguintes informações:

Entidade: É o nome da tabela no Banco de Dados;

Atributo: São as características da entidade armazenadas no banco;

Tipo de Dado: São os tipos primitivos que o atributo armazena no banco de dados;

PK: Primary Key – Chave Primária;

FK: Foreign Key – Chave Secundária;

NN: Not Null – Não Nulo;

UN: Unique – Único;

AI: Auto Increment – Auto Incrementável;

Restrições: Descreve as restrições que o atributo possa ter.

Quadro 3 - Dicionário de dados da entidade avisos

Entidade: avisos							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
titulo	VARCHAR(70)			✓			
mensagem	VARCHAR(550)			✓			
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 4 - Dicionário de dados da entidade cautelas

Entidade: cautelas							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
idArmeiro	BIGINT		✓	✓	✓		
idPolicial	BIGINT		✓	✓	✓		
idMaterial	BIGINT		✓	✓	✓		
observacao	VARCHAR(191)						NULL
devolucao	DATETIME						NULL
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 5 - Dicionário de dados da entidade failed_jobs

Entidade: failed_jobs							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
uuid	VARCHAR(191)			✓			
connection	TEXT			✓			
queue	TEXT			✓			
payload	LONGTEXT			✓			
exception	LONGTEXT			✓			
failed_at	TIMESTAMP			✓			CURRENT_TIMESTAMP

Fonte: Elaborado pelo autor (2023)

Quadro 6 - Dicionário de dados da entidade materials

Entidade: materials							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
idTipoMaterial	BIGINT		✓	✓	✓		
tombamento	VARCHAR(191)			✓			
nroSerie	VARCHAR(191)			✓			
marca	VARCHAR(191)			✓			
modelo	VARCHAR(191)			✓			
calibre	VARCHAR(191)						NULL
tamanho	VARCHAR(191)						NULL
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 7 - Dicionário de dados da entidade migrations

Entidade: migrations							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	INT	✓		✓	✓	✓	
migration	VARCHAR(191)			✓			
batch	INT			✓			

Fonte: Elaborado pelo autor (2023)

Quadro 8 - Dicionário de dados da entidade model_has_permissions

Entidade: model_has_permissions							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
permission_id	BIGINT	✓	✓	✓	✓		
model_type	VARCHAR(191)	✓		✓			
model_id	BIGINT	✓		✓	✓		

Fonte: Elaborado pelo autor (2023)

Quadro 9 - Dicionário de dados da entidade model_has_holes

Entidade: model_has_roles							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
role_id	BIGINT	✓	✓	✓	✓		
model_type	VARCHAR(191)	✓		✓			
model_id	BIGINT	✓		✓	✓		

Fonte: Elaborado pelo autor (2023)

Quadro 10 - Dicionário de dados da entidade password_reset_tokens

Entidade: password_reset_tokens							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
email	VARCHAR(191)	✓		✓			
token	VARCHAR(191)			✓			
created_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 11 - Dicionário de dados da entidade permissions

Entidade: permissions							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
name	VARCHAR(191)			✓			
guard_name	VARCHAR(191)			✓			
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 12 - Dicionário de dados da entidade personal_access_tokens

Entidade: personal_access_tokens							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
tokenable_type	VARCHAR(191)			✓			
tokenable_id	BIGINT			✓	✓		
name	VARCHAR(191)			✓			
token	VARCHAR(64)			✓			
abilities	TEXT						NULL
last_used_at	TIMESTAMP						NULL
expires_at	TIMESTAMP						NULL
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 13 - Dicionário de dados da entidade policials

Entidade: policials							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
rePolicial	INT			✓			
nome	VARCHAR(191)			✓			
nomeGuerra	VARCHAR(191)			✓			
senha	INT						NULL
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 14 - Dicionário de dados da entidade role_has_permissions

Entidade: role_has_permissions							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
permission_id	BIGINT	✓	✓	✓	✓		
role_id	BIGINT	✓	✓	✓	✓		

Fonte: Elaborado pelo autor (2023)

Quadro 15 - Dicionário de dados da entidade roles

Entidade: roles							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
name	VARCHAR(191)			✓			
guard_name	VARCHAR(191)			✓			
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 16 - Dicionário de dados da entidade tipo_materials

Entidade: tipo_materials							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
tipoMaterial	VARCHAR(191)			✓			
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)

Quadro 17 - Dicionário de dados da entidade users

Entidade: users							
Atributo	**Tipo de Dado**	**PK**	**FK**	**NN**	**UN**	**AI**	**Restrições**
id	BIGINT	✓		✓	✓	✓	
name	VARCHAR(191)			✓			
email	VARCHAR(191)			✓			
email_verified_at	TIMESTAMP						NULL
password	VARCHAR(191)			✓			
remember_token	VARCHAR(100)						NULL
created_at	TIMESTAMP						NULL
updated_at	TIMESTAMP						NULL

Fonte: Elaborado pelo autor (2023)