

Erik Barcella Trisch

**AutoForm- Sistema para o Registro de Produtos
Controlados no SIGMA via Arquivo Eletrônico em
Lote (AEL) da Brigada Militar do RS**

Osório

2023

Erik Barcella Trisch

**AutoForm- Sistema para o Registro de Produtos Controlados
no SIGMA via Arquivo Eletrônico em Lote (AEL) da Brigada
Militar do RS**

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do título
de Tecnólogo em Análise e Desenvolvimento de
Sistemas.

Orientador: Bruno Chagas Fernandes

Coorientador: Márcio José de Lemos

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS

Campus Osório

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Osório

2023

Erik Barcella Trisch

AutoForm- Sistema para o Registro de Produtos Controlados no SIGMA via Arquivo Eletrônico em Lote (AEL) da Brigada Militar do RS

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Bruno Chagas Fernandes
Orientador

Márcio José de Lemos
Coorientador

Tiago Guimarães Moraes
Convidado 1

Marcelo Paravisi
Convidado 2

Osório
2023

Este trabalho é dedicado para minha esposa Vitória que esteve ao meu lado, oferecendo apoio incondicional durante o desenvolvimento desse projeto. Muito obrigado por compartilhar esse caminho comigo.

AGRADECIMENTOS

Os agradecimentos principais são destinados à minha esposa Vitória, e à minha família, que estiveram sempre ao meu lado, proporcionando apoio e incentivo ao longo desta jornada. Vocês são a minha base, as fontes constantes de inspiração que possibilitaram alcançar este objetivo.

Agradeço imensamente ao Sd. Tiago Costa dos Santos por ser o elo de idealização deste trabalho junto à Brigada Militar, fornecendo apoio e suporte essenciais para a conclusão deste projeto. Sua colaboração foi fundamental para a realização deste trabalho.

Aos meus dedicados orientador e coorientador, Bruno Chagas Alves Fernandes e Márcio José de Lemos, expresso profundos agradecimentos. Suas orientações foram cruciais, e a dedicação, paciência e contribuições de ambos enriqueceram significativamente este trabalho. Agradeço pelo comprometimento e pelos valiosos ensinamentos que foram essenciais para o sucesso deste projeto acadêmico.

Agradeço também aos demais que contribuíram de alguma forma para que isso fosse possível.

”Só se pode alcançar um grande êxito quando nos mantemos fiéis a nós mesmos”
Friedrich Nietzsche

RESUMO

A Brigada Militar do Rio Grande do Sul é responsável por gerar um documento eletrônico denominado AEL que inclui dados sobre as armas registradas no estado, e encaminhar à Diretoria de Fiscalização de Produtos Controlados do Exército para cadastro no Sistema de Gerenciamento Militar de Armas (SIGMA). Em razão da demanda apresentada pela BM RS por um sistema que sustente a execução deste processo, foi sugerido o desenvolvimento desta aplicação web, denominada AutoForm, desenvolvida com a linguagem JavaScript em conjunto com os frameworks React e NodeJS. Utilizando-se de estratégias e metodologias que serão abordados durante esta pesquisa, visando facilitar o preenchimento das informações pelo operador, otimizar o tempo de execução desta tarefa, aumentar a eficácia, e contemplar todos os requisitos necessários para geração do AEL garantindo que este esteja completo e correto antes de ser submetido ao SIGMA.

Palavras-chave: Arquivo Eletrônico em Lote, SIGMA, Brigada Militar, Aplicação Web, React, NodeJS, JavaScript.

ABSTRACT

The Rio Grande do Sul Military Brigade is responsible for generating an electronic document called AEL that includes data on weapons registered in the state, and forwarding it to the Army's Controlled Products Inspection Directorate for registration in the Military Weapons Management System (SIGMA). Due to the demand presented by BM RS for a system that supports the execution of this process, the development of this web application, called AutoForm, developed with the JavaScript language was suggested in conjunction with the React and NodeJS frameworks. Using strategies and methodologies that will be addressed during this research, we aim to facilitate the completion of information by the operator, optimize the execution time of this task, increase efficiency, and take into account all necessary requirements for generating the AEL, ensuring that it is complete and correct before being submitted to SIGMA.

Keywords: Electronic Batch File, SIGMA, Military Brigade, Web Application, React, NodeJS, JavaScript.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura Cliente-Servidor	19
Figura 2 – Arquitetura MVC	20
Figura 3 – Fluxo de autenticação JWT	25
Figura 4 – Tela de cadastros da aplicação CRPG	26
Figura 5 – Tela de cadastros da aplicação de Gerenciamento de Licenças	27
Figura 6 – Tela Inicial do aplicativo TAF BM	28
Figura 7 – Disposição das tecnologias na arquitetura MVC	37
Figura 8 – Diagrama da estrutura de armas	38
Figura 9 – Diagrama da estrutura de usuário	38
Figura 10 – Autoform - Página de login	39
Figura 11 – Autoform - Página de registros	40
Figura 12 – Autoform - Página inicial home	41
Figura 13 – Autoform - Página de criação do AEL	42
Figura 14 – Autoform - Página de criação do AEL com opção para selecionar	42
Figura 15 – Autoform - Página de criação do AEL -2	43
Figura 16 – Autoform - Página de criação do AEL-3	43
Figura 17 – Autoform - Página de criação do AEL-4	44
Figura 18 – Autoform - AEL gerado	45
Figura 19 – Autoform - Cadastros	46
Figura 20 – Autoform - Administração de usuários	47

LISTA DE TABELAS

Tabela 1 – Comparação entre SIGMA e SINARM	16
Tabela 2 – Comparação Entre Bibliotecas e Frameworks no Desenvolvimento de Software	22
Tabela 3 – Requisito Funcional 1	30
Tabela 4 – Requisito Funcional 2	30
Tabela 5 – Requisito Funcional 3	31
Tabela 6 – Requisito Funcional 4	31
Tabela 7 – Requisito Funcional 5	31
Tabela 8 – Requisito Funcional 6	31
Tabela 9 – Requisito Funcional 7	32
Tabela 10 – Requisito Funcional 8	32
Tabela 11 – Requisito Funcional 9	32
Tabela 12 – Requisito Funcional 10	32
Tabela 13 – Requisito Funcional 11	33
Tabela 14 – Requisito Funcional 12	33
Tabela 15 – Requisito Funcional 13	33
Tabela 16 – Requisito Funcional 14	34
Tabela 17 – Requisito Não Funcional 1	34
Tabela 18 – Requisito Não Funcional 2	34
Tabela 19 – Requisito Não Funcional 3	34
Tabela 20 – Requisito Não Funcional 4	35
Tabela 21 – Requisito Não Funcional 5	35
Tabela 22 – Requisito Não Funcional 6	35
Tabela 23 – Requisito Não Funcional 7	35

LISTA DE ABREVIATURAS E SIGLAS

SIGMA	Sistema de Gerenciamento Militar de Armas
AEL	Arquivos Eletrônicos em Lote
SINARM	Sistema Nacional de Armas
QG	Quartel General
OM	Organizações Militares
MVC	<i>Model-View-Controller</i>
HTTP	<i>Hypertext Transfer Protocol</i>
API	<i>Application Programming Interface</i>
REST	<i>Representational State Transfer</i>
JSON	<i>JavaScript Object Notation</i>
BM RS	Briga Militar do Rio Grande do Sul
UI	<i>User Interface</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Justificativa	14
1.2	Objetivos	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos específicos	14
2	REFERENCIAL TEÓRICO	16
2.1	Controle na Segurança Pública Brasileira	16
2.1.1	SIGMA e SINARM	16
2.2	Sistema SIGMA	16
2.2.1	Contexto de implantação	17
2.2.2	AEL	17
2.2.3	Arquivo AEL na Brigada Militar do Rio Grande do Sul	17
2.3	Gerenciamento de processos	17
2.3.1	BPMN	18
2.4	Padrões de Projeto de Sistemas	18
2.4.1	Arquitetura Cliente Servidor	18
2.4.2	Arquitetura MVC	19
2.5	Aplicações Web	20
2.5.1	Linguagem JavaScript	21
2.5.2	Frameworks e Bibliotecas	21
2.5.2.1	Node	22
2.5.2.2	ReactJS	23
2.6	Banco de dados NOSQL	23
2.6.1	Segurança e autenticação	24
3	TRABALHOS RELACIONADOS	26
3.1	Aplicação WEB Para Pessoas Físicas Que Utilizam Produtos Controlados Pelo Exército Brasileiro E Polícia Federal	26
3.2	Sistema De Gerenciamento De Licenças De Posse E Porte De Armas De Fogo	27
3.3	TAF- Teste de Aptidão Física da Brigada Militar do Rio Grande do Sul	27
4	METODOLOGIA	29
4.1	Estruturação do desenvolvimento	29
4.2	Análise de Requisitos	30

4.2.1	Requisitos Funcionais	30
4.2.2	Requisitos Não Funcionais	34
5	DESENVOLVIMENTO DA APLICAÇÃO AUTOFORM	36
5.1	Processo de produção	36
5.1.1	Tecnologias	36
5.1.2	Arquitetura da aplicação	37
5.1.3	Aplicação Web AutoForm	38
5.1.4	Autenticação	38
5.1.5	Página de login	39
5.1.6	Página de registro	40
5.1.7	Página inicial	40
5.1.8	Página de criação de AEL	41
5.1.9	Página de criação de AEL com opção de autopreenchimento	41
5.1.10	Exemplo de AEL gerado	45
5.1.11	Página de cadastros	45
5.1.12	Página de administração de usuários	46
6	CONCLUSÃO	48
6.0.1	Resultados finais	48
6.1	Trabalhos futuros	48
	REFERÊNCIAS	50
	ANEXO A – PORTARIA 136 - ANEXO D.1	52
	ANEXO B – PORTARIA 136 - ANEXO D.2	53
	ANEXO C – PORTARIA 136 - ANEXO D.3	54
	ANEXO D – PORTARIA 136 - ANEXO D.4	55

1 INTRODUÇÃO

O Sistema de Gerenciamento Militar de Armas (SIGMA) é um sistema informatizado utilizado como ferramenta de controle e rastreamento para gerenciar o registro e transferência de armas de fogo, munições e demais produtos controlados de competência do Comando do Exército em todo o território Brasileiro ([EXÉRCITO BRASILEIRO, 2019](#)).

Sua criação e implementação foram conduzidas pelo Ministério da Defesa, em coordenação com o Comando do Exército. O SIGMA tem a finalidade de administrar os registros de armas de propriedade particular pertencentes a diversos grupos, incluindo as armas de fogo de integrantes das Forças Armadas, das Polícias Militares e dos órgãos de segurança pública, além de empresas de segurança privada e colecionadores de armas. Em essência, o SIGMA atua como um sistema centralizado de gerenciamento de informações sobre armas de fogo sob posse dessas entidades e indivíduos, contribuindo para a regulamentação e controle no contexto nacional ([EXÉRCITO BRASILEIRO, 2019](#)).

Na Brigada Militar do Rio Grande do Sul, o setor administrativo de cadastros e registros é responsável por controlar e manter diversos processos que abrangem uma variedade de assuntos relacionados às atividades e operações da instituição militar, inclusive interligados a outros órgãos públicos ([BMRS, 2023](#)).

Portanto um dos processos administrativos mantidos pela corporação é a geração do AEL, onde atualmente esta tarefa é realizada pelos seus operadores de forma manual preenchendo os dados necessários em um arquivo de texto, seguido da formatação e adequação do documento ao modelo padrão estabelecido pelo Exército Brasileiro, que possui regras específicas de indexação das informações conforme especificado no manual de preenchimento do arquivo eletrônico (anexo [A](#)).

Após a geração do arquivo eletrônico, o mesmo é submetido para avaliação da Diretoria de Fiscalização de Produtos Controlados do Exército, que irá verificar as informações contidas no arquivo e fazer o *upload* para o SIGMA, após validação do sistema, em caso de sucesso do cadastro, é retornado um número de identificação único gerado pelo SIGMA, o qual é utilizado pelo exército para identificar o respectivo registro e todas suas informações dentro do SIGMA quando necessário ([EXÉRCITO BRASILEIRO, 2019](#)).

E, utilizado pela Brigada Militar o número de identificação gerado pelo SIGMA como retorno de que o cadastro daquele processo foi efetivado na base de dados e finalizado, conforme evidenciado seção 3.4 alínea C do anexo [D](#).

Entretanto, no momento atual o AEL é formatado manualmente pelo operador após o preenchimento, implicando em uma maior complexidade na execução dessa tarefa. A proposta deste trabalho é desenvolver uma aplicação web que automatize o processo de geração do arquivo

eletrônico visando simplificar significativamente o procedimento, através do preenchimento facilitado proporcionando resultados mais eficazes e alavancando a excelência operacional da instituição.

1.1 JUSTIFICATIVA

A divisão interna da BM RS desempenha o papel crucial de supervisionar e gerir uma série de procedimentos administrativos que abrangem uma ampla gama de assuntos relacionados às atividades e operações da instituição militar (BMRS, 2023).

Uma das atividades administrativas sob a responsabilidade da corporação é registrar e manter atualizado o cadastro, transferência de armas de fogo e demais produtos regulados sob a jurisdição do Comando do Exército no estado do Rio Grande do Sul, através do envio de documento eletrônico para a DFPC destinado a registrar no SIGMA (EXÉRCITO BRASILEIRO, 2019).

Neste momento, a execução dessa atividade é realizada de forma manual por operadores da BM RS, os quais inserem os dados essenciais em um documento de texto. Além, de prece-derem com a formatação do documento de acordo com as diretrizes estipuladas pelo Exército Brasileiro, que estabelece regras específicas para a indexação das informações, conforme minuciosamente delineado no manual de preenchimento de arquivo eletrônico (anexo D).

Nesse sentido, foi estipulado o desenvolvimento uma aplicação digital a fim de simplifi-car notavelmente o processo de preenchimento e geração do AEL na Brigada Militar, de forma que esta plataforma contribua de maneira eficiente e decisiva.

1.2 OBJETIVOS

Os objetivos esperados são apresentados nesta seção abaixo, divididos em geral e espe-cíficos.

1.2.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma aplicação web que contribua de forma eficaz para o preenchimento e geração do AEL, buscando otimizar o fluxo do processo para a Brigada Militar, através de preenchimento facilitado e geração automática do arquivo conforme o padrão pré estabelecido pelo anexo D da portaria 136 de 8 de novembro de 2019 do Exército Brasileiro (anexo D).

1.2.2 Objetivos específicos

- Realizar o levantamento de requisitos necessários para o desenvolvimento da aplicação web.

- Desenvolver o cadastro de armas, persistindo no banco de dados.
- Autopreencher os campos do formulário com informações da arma quando selecionada uma previamente cadastrada pelo operador.
- Permitir o gerenciamento de usuários através de um acesso especial com permissão de administrador do sistema.
- Facilitar o fluxo para o operador no preenchimento do formulário web, retornando para o usuário possíveis erros e regras específicas que devem ser seguidas.
- Automatizar a geração do arquivo.

2 REFERENCIAL TEÓRICO

Neste capítulo é abordado o referencial teórico, utilizado como embasamento para a construção deste trabalho.

2.1 CONTROLE NA SEGURANÇA PÚBLICA BRASILEIRA

Atualmente no Brasil existem dois sistemas informatizados utilizados por órgãos públicos para realizar a regulamentação e o monitoramento de armas de fogo, munições e demais produtos controlados. Sendo estes, o Sistema de Gerenciamento Militar de Armas (SIGMA) e o Sistema Nacional de Armas (SINARM). O SIGMA é administrado pelo Exército Brasileiro e é responsável pelo controle de armas de fogo e munições no âmbito da Força. O SINARM é administrado pela Polícia Federal e é responsável pelo controle de armas de fogo e munições em poder da população civil (CARADA, 2019).

2.1.1 SIGMA e SINARM

A principal diferença entre SIGMA e SINARM é o âmbito de atuação. O SIGMA é responsável pelo controle de armas de fogo e munições no âmbito do Exército Brasileiro, enquanto o SINARM é responsável pelo controle de armas de fogo e munições em poder da população civil. Outra diferença entre os dois sistemas é a natureza das informações que eles gerenciam. O SIGMA gerencia informações sobre armas de fogo e munições de uso militar, enquanto o SINARM gerencia informações sobre armas de fogo e munições de uso civil. A seguir, está uma tabela comparativa que resume as principais diferenças entre ambos:

Tabela 1 – Comparação entre SIGMA e SINARM

Característica	SIGMA	SINARM
Âmbito de atuação	Exército Brasileiro	População civil
Natureza das informações	Armas de fogo e munições de uso militar	Armas de fogo e munições de uso civil
Responsável pela administração	Exército Brasileiro	Polícia Federal

2.2 SISTEMA SIGMA

O Sistema de Gerenciamento Militar de Armas (SIGMA) é um sistema computacional desenvolvido pelo Centro de Desenvolvimento de Sistemas (CDS) do Exército Brasileiro e implantado em 2003 que vem sendo constantemente atualizado para atender às necessidades da força (FENEME, 2007).

2.2.1 Contexto de implantação

O contexto da implantação do SIGMA foi a necessidade de modernizar o sistema de controle de armas de fogo e munições do Exército Brasileiro. O SIGMA foi desenvolvido com base nas melhores práticas internacionais de controle de armas de fogo. O sistema é integrado a outros sistemas de informação do Exército Brasileiro, o que permite a troca de dados e informações entre as diferentes áreas da força (FENEME, 2007).

2.2.2 AEL

O Arquivo Eletrônico em Lote (AEL) é um arquivo digital que contém as informações necessárias para o cadastro de produtos controlado no SIGMA. O AEL é utilizado para o cadastro de armas de fogo de diversas entidades, como as Forças Armadas, as forças auxiliares, a Polícia Militar e o Corpo de Bombeiros (EXÉRCITO BRASILEIRO, 2019).

O Objetivo do AEL no sistema SIGMA é permitir o cadastro de produtos controlados de diversas entidades de forma centralizada e organizada. O AEL é um elemento importante do SIGMA, pois permite que o Exército Brasileiro tenha um controle mais eficiente das armas de fogo em circulação no país (EXÉRCITO BRASILEIRO, 2019).

2.2.3 Arquivo AEL na Brigada Militar do Rio Grande do Sul

O AEL no contexto da BM RS, deve ser gerado para o cadastro de armas de fogo de policiais militares. O arquivo deve conter as seguintes informações:

- Identificação da Brigada Militar: número do QG, código da OM e nome da OM.
- Identificação do armamento: número da arma, tipo de arma, marca, modelo, calibre e série.
- Identificação do proprietário: nome completo, CPF, RG, endereço e telefone.
- Além das demais informações especificadas nos anexos B e C.

O AEL deve ser gerado em um formato texto, seguindo um layout pré-definido e estar conforme os parâmetros de indexação das informações constantes nos anexos A e D.

2.3 GERENCIAMENTO DE PROCESSOS

É uma abordagem disciplinada e sistemática que envolve práticas relacionadas aos processos de negócio, automatizados ou não, com o objetivo de alcançar resultados consistentes e alinhados com as metas estratégicas de uma organização. Conforme DÁVILA, Leocádio e Varvakis (2008) “As organizações tentam inovar para se diferenciar e obter vantagens competitivas,

tanto pela melhoria nos bens/serviços fornecidos quanto pela eficiência operativa”. Pode-se concluir que os sistemas de informação oferecem inúmeros benefícios para uma organização, sejam eles para melhorar o fluxo de informação, as tomadas de decisões, o controle de qualidade, ou ampliar a produtividade.

2.3.1 BPMN

O modelo e notação de processos de negócios (*Business Process Model and Notation*) é uma notação gráfica padronizada para desenhar processos de negócios em um fluxograma. A diagramação BPMN é intuitiva e permite a representação de detalhes complexos do processo. A simbologia deste modelo serve como uma linguagem padrão, colocando um fim na lacuna de comunicação entre a modelagem do processo e sua execução, para [Bitencourt, Paiva e Cagnin \(2016\)](#):

Modelo de processos de negócio representa os processos de negócio de uma empresa e permite a documentação, simulação, compartilhamento, implementação, avaliação e melhoramento contínuo das operações, com o intuito de compreender o funcionamento da organização e os aspectos do seu domínio ([BITENCOURT; PAIVA; CAGNIN, 2016](#)).

Em resumo, o levantamento e registro da situação atual dos processos, seguido por uma análise aprofundada, são práticas essenciais para promover a eficiência, a eficácia e a adaptação contínua dentro de uma organização. Essa abordagem sistemática para entender e aprimorar os processos é fundamental para a sustentabilidade organizacional.

2.4 PADRÕES DE PROJETO DE SISTEMAS

Padrões de desenvolvimento de software referem-se a soluções reutilizáveis para problemas comuns encontrados no processo de desenvolvimento de software. Esses padrões são abstrações que encapsulam as melhores práticas, representando soluções testadas e comprovadas para desafios recorrentes. Eles fornecem diretrizes para o design e implementação de código, promovendo a consistência, a manutenibilidade e a eficiência no desenvolvimento de software ([FIGUEIREDO, 2014](#)).

O contexto dos padrões de desenvolvimento de software está relacionado aos desafios enfrentados pelos desenvolvedores ao criar sistemas de software complexos. Conforme abordado por [Figueiredo \(2014\)](#) “Um padrão de projeto nomeia, abstrai e identifica aspectos problemáticos comuns e propõe uma solução padrão para esses problemas”.

2.4.1 Arquitetura Cliente Servidor

Nessa arquitetura, o software é dividido em duas partes principais: o cliente e o servidor.

O cliente é a parte do sistema que interage diretamente com o usuário. Ele envia solicitações de serviço ao servidor e exibe os resultados recebidos ao usuário. O cliente pode ser um

aplicativo de desktop, um aplicativo móvel ou um navegador da web, dependendo do tipo de sistema que está sendo desenvolvido (FLANAGAN, 2012).

O servidor é responsável por processar as solicitações recebidas do cliente e fornecer os recursos ou serviços solicitados. Ele possui os recursos necessários para atender às solicitações, como bancos de dados, aplicativos e serviços web. O servidor está sempre ativo, aguardando solicitações dos clientes e respondendo a elas de maneira apropriada (OLIVEIRA, 2003).

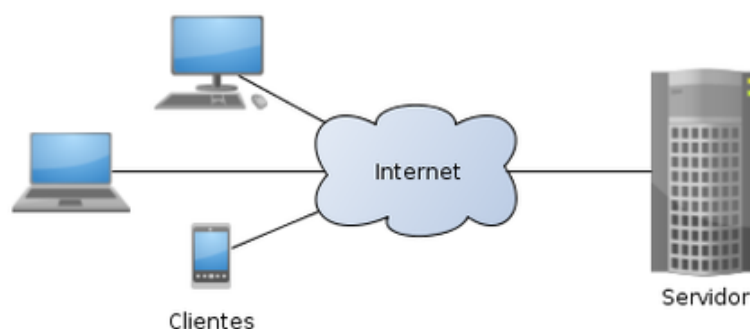
A comunicação entre o cliente e o servidor ocorre por meio de uma rede, geralmente a Internet. O cliente envia uma solicitação para o servidor, especificando o tipo de serviço desejado e quaisquer parâmetros necessários. Então de acordo com Oliveira (2003).

O servidor, quando recebe a mensagem, extrai os parâmetros e chama o procedimento especificado na mensagem. No fim da execução do procedimento é realizada a operação inversa, colocando os resultados e enviando a mensagem de resposta ao processo cliente (OLIVEIRA, 2003).

Logo uma das principais vantagens da arquitetura cliente-servidor é a divisão clara de responsabilidades entre ambos. O cliente lida com a interface do usuário e a apresentação dos dados, enquanto o servidor cuida do processamento das solicitações e do acesso aos recursos. Isso permite uma melhor organização do sistema e facilita a manutenção e a escalabilidade. Além disso, a arquitetura cliente-servidor permite que vários clientes acessem o mesmo servidor simultaneamente. Isso possibilita o compartilhamento de recursos e serviços, o que é especialmente útil em ambientes corporativos (OLIVEIRA, 2003).

A Figura 1 demonstra a maneira como ocorre essa comunicação:

Figura 1 – Arquitetura Cliente-Servidor



Fonte: Redes (2016)

2.4.2 Arquitetura MVC

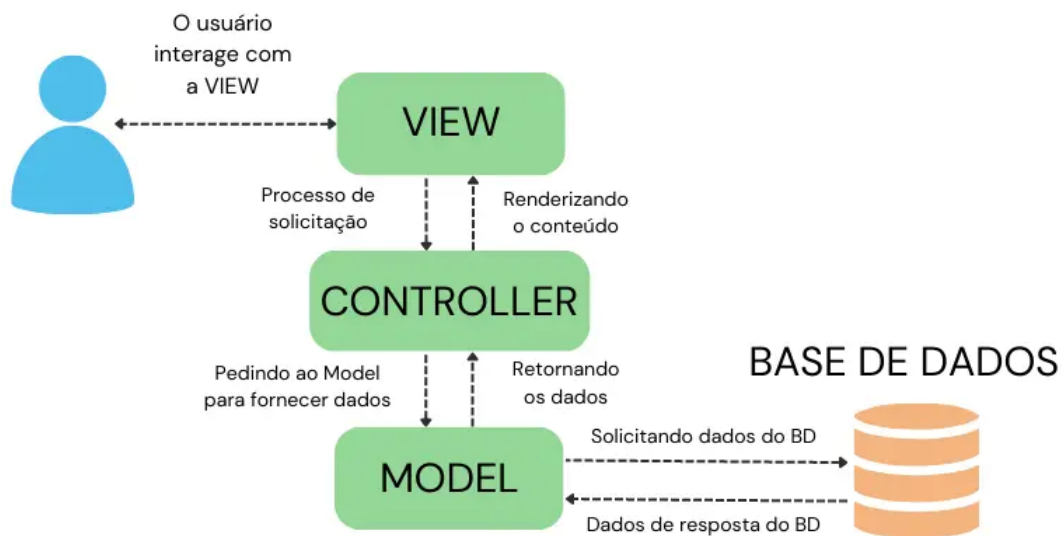
A arquitetura MVC *Model-View-Controller* é um padrão de design que organiza o código de uma aplicação em três componentes principais: *Model* (Modelo), *View* (Visão) e *Controller*

(Controlador). Cada componente tem uma responsabilidade específica na aplicação, o que ajuda a manter o código modular, escalável e de fácil manutenção (VALENTE, 2020).

- Visão: Lida com a apresentação dos dados ao usuário e interage com o Modelo. A Visão exibe as informações e envia eventos do usuário para o Controlador.
- Controlador: Recebe entradas do usuário, processa essas entradas (geralmente envolvendo o Modelo) e atualiza a Visão. O Controlador age como um intermediário entre o Modelo e a Visão.
- Modelo: Representa a lógica de negócios e os dados da aplicação. Geralmente, o modelo é responsável pela interação com o banco de dados e pela manipulação dos dados.

Conforme exemplificado o fluxo da arquitetura MVC na Figura 2

Figura 2 – Arquitetura MVC



Fonte: Valente (2020)

2.5 APLICAÇÕES WEB

Aplicações web são programas que são executados em navegadores e são acessados por meio de internet. Surgiram na década de 1990 e se tornaram populares por permitirem a interação do usuário e o processamento de dados.

Existem dois tipos principais: estáticas (HTML, CSS e JavaScript) e dinâmicas (linguagens de programação do lado do servidor). De acordo com GALHARDI (2009) as aplicações estáticas geralmente consistem em páginas web com conteúdo fixo, sem interação avançada ou

processamento de dados em tempo real. O navegador do cliente solicita páginas estáticas ao servidor, que retorna arquivos HTML, CSS e JavaScript. A renderização e interação ocorrem no navegador.

Aplicações Dinâmicas apresentam interatividade avançada e processamento de dados em tempo real quando o navegador solicita uma página ao servidor. O servidor executa a lógica de negócios, acessa dados do banco de dados, gera dinamicamente o conteúdo HTML e o envia de volta ao navegador. Pode haver interações adicionais entre o navegador e o servidor oferecendo diversos benefícios como acessibilidade, atualização, redução de custos e escalabilidade (GALHARDI, 2009).

2.5.1 Linguagem JavaScript

Uma linguagem de programação amplamente usada no desenvolvimento web de acordo com a organização MDN (2022). Ela permite adicionar interatividade e dinamismo a páginas da web. Além de ser usado no desenvolvimento de interfaces de usuário, o JavaScript também pode ser usado no desenvolvimento de aplicativos do lado do servidor (backend) com o uso de tecnologias como o Node.js. Conforme Flanagan (2012) “Javascript já deixou para trás suas raízes como linguagem de script há muito tempo, tornando-se uma linguagem de uso geral, robusta e eficiente”.

2.5.2 Frameworks e Bibliotecas

A necessidade da utilização de *frameworks* surgiu com a complexidade crescente das aplicações de software.

Diversos *frameworks* têm sido desenvolvidos nas duas últimas décadas, visando o reuso de software e consequentemente a melhoria da produtividade, qualidade e manutenibilidade (MALDONADO et al., 2002).

Sendo estas estruturas ou conjuntos de ferramentas que fornecem uma base organizada para o desenvolvimento de aplicações. Eles oferecem uma estrutura pré-definida que acelera o processo de desenvolvimento, promove a reutilização de código e estabelece padrões de boas práticas. No contexto do desenvolvimento de software, os frameworks desempenham um papel significativo, influenciando a forma como as aplicações são projetadas, implementadas e mantidas (MALDONADO et al., 2002).

Portanto para Maldonado et al. (2002, p. 23) os *frameworks* estão dividido em dois tipos sendo eles:

- Caixa preta: Uma abordagem caixa preta trata um sistema ou componente como uma entidade onde o foco está no comportamento externo, sem conhecimento detalhado de sua implementação interna, pois na maioria dos casos o desenvolvedor não tem acesso ao código fonte e a ênfase está nos resultados visíveis e nas funcionalidades oferecidas pelo

sistema, sem a necessidade de compreender a lógica interna do sistema (MALDONADO et al., 2002, p. 23).

- Caixa branca: Em contraste, uma abordagem caixa branca envolve uma compreensão detalhada da implementação interna de um sistema ou componente, incluindo sua lógica, estrutura e fluxo de controle e a ênfase está na compreensão completa do sistema, possibilitando otimizações, depuração precisa e ajustes finos (MALDONADO et al., 2002, p. 23).

Entretanto muito semelhante aos *frameworks* exceto por algumas diferenças abordados na Tabela 2, existem também as bibliotecas de *software* que são conjuntos de códigos preexistentes e funcionalidades encapsuladas que foram desenvolvidos para abordar tarefas comuns ao desenvolvimento de sistemas. Conforme Cechinel et al. (2017) torna-se evidente que esses recursos desempenham um papel fundamental na eficiência e na evolução contínua do desenvolvimento de *software*. Desde a reutilização inteligente de código até a adaptação às inovações tecnológicas.

Característica	Bibliotecas	Frameworks
Definição	Fornecem funcionalidades específicas.	Oferecem uma estrutura abrangente.
Flexibilidade	Mais flexibilidade para os desenvolvedores.	Menos flexibilidade devido a maiores convenções.
Integração	Integração opcional e modular.	Impõem uma estrutura mais integrada.
Modularidade	Modular, escolha de partes específicas.	Estrutura monolítica, menos modularidade.

Tabela 2 – Comparação Entre Bibliotecas e Frameworks no Desenvolvimento de Software

2.5.2.1 Node

Conforme o site oficial do Node (2009), este é um ambiente de tempo de execução javascript que permite que o javascript seja executado no lado do servidor. Utilizando o mecanismo V8 do Google Chrome para executar código javascript fora do navegador. Com o Node.js, é possível criar aplicativos web e serviços *backend* usando javascript. Ele fornece uma variedade de recursos e uma ampla gama de bibliotecas e frameworks, tornando-o uma escolha popular para o desenvolvimento de servidores e APIs (NODE, 2009). Portanto para Pereira (2014).

Node.js é multiprotocolo, ou seja, com ele será possível trabalhar com os protocolos: HTTP, HTTPS, FTP, SSH, DNS, TCP, UDP, WebSockets e também existem outros. Toda aplicação web necessita de um servidor para disponibilizar todos os seus recursos (PEREIRA, 2014).

Amplamente utilizado em conjunto com o node, é o framework express para aplicativos web do lado do servidor, construído sob a base nativa HTTP do Node.js. Ele fornece uma abordagem simplificada para lidar com solicitações HTTP, roteamento e manipulação de middleware. O Express permite criar facilmente APIs robustas e eficientes, tornando o desenvolvimento de aplicativos web mais rápido e produtivo. É um dos frameworks mais populares para o desenvolvimento de servidores com Node.js (PEREIRA, 2014).

2.5.2.2 ReactJS

React é uma biblioteca javascript *Open Source* lançado em 2013 pela [Meta Open Source \(2013\)](#), que rapidamente ganhou popularidade devido a sua abordagem inovadora utilizada para criar interfaces de usuário. Através da possibilidade de escrever código utilizando a sintaxe JSX, a qual é uma convenção opcional no react que possibilita ter a organização do código javascript de maneira mais próxima à estrutura de marcação XML ou HTML.

Logo essa biblioteca permite desenvolver componentes reutilizáveis e interativos utilizados na construção de *UIs* modernas e responsivas. Possibilitando o desenvolvimento de aplicações web complexas e dinâmicas, dividindo-as em componentes reutilizáveis [Meta Open Source \(2013\)](#).

A partir da premissa de que a abordagem principal do React é baseada em componentes, isso facilita a criação e o gerenciamento de estado dos elementos da interface. Permitindo a criação de aplicações eficientes e escaláveis, sendo como principal característica a renderização reativa, o que faz com que a interface do usuário seja atualizada automaticamente quando o estado dos dados é alterado. Isso simplifica o desenvolvimento e melhora o desempenho, pois apenas as partes afetadas da interface são atualizadas, em vez de recarregar a página inteira ([Meta Open Source, 2013](#)).

2.6 BANCO DE DADOS NOSQL

Banco de dados NoSQL é um tipo de banco de dados que difere dos bancos de dados relacionais tradicionais (SQL) em sua estrutura de armazenamento e modelo de dados. NoSQL significa *Not Only SQL* (Não Apenas SQL) e abrange diversos tipos de bancos de dados que oferecem uma abordagem alternativa para o armazenamento e recuperação de dados. Para [Pereira \(2014\)](#) uma das vantagens em se trabalhar com um banco de dados desse modelo é o grande suporte oferecido pela comunidade do nodejs e uma vasta gama de compatibilidade com diversas tecnologias.

Dentre as principais tecnologias que tem uma alta sinergia nesse padrão não relacional são:

- MongoDB: um eficiente e popular banco de dados NoSQL. Que utiliza um sistema de gerenciamento de banco de dados orientado a documentos, o que significa que os dados

são armazenados em documentos semelhantes a JSON, em vez de tabelas com linhas e colunas como em um banco de dados relacional (PEREIRA, 2014). Outra característica importante do MongoDB é sua capacidade de escalar horizontalmente. Oferecendo recursos avançados, como indexação, consultas poderosas e suporte a transações, tornando-o adequado para uma ampla gama de aplicações. É frequentemente utilizado em aplicativos web, análise de dados, e outras aplicações que exigem flexibilidade e escalabilidade (PEREIRA, 2014).

- Mongoose: Uma biblioteca ODM *Object Data Modeling* para Node.js e MongoDB. Sendo inserida como uma camada de abstração facilitando a conexão, a modelagem de dados, a execução de consultas, e a interação com o banco de dados de maneira eficiente e organizada (PEREIRA, 2014).

2.6.1 Segurança e autenticação

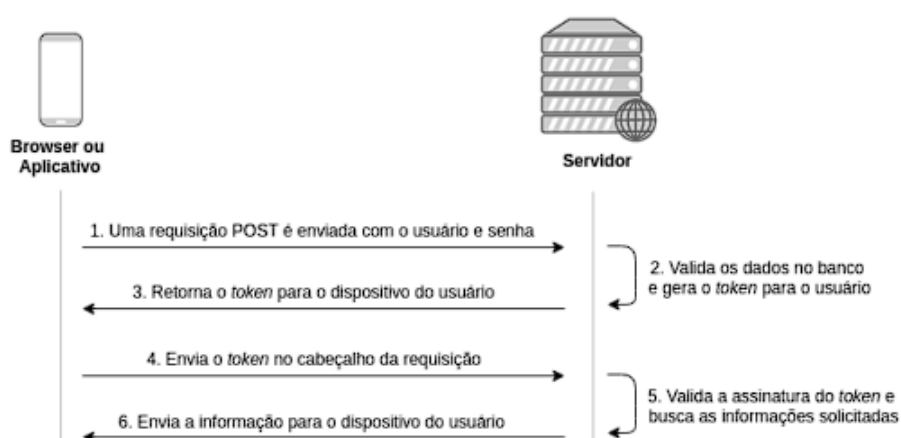
A segurança e a autenticação em aplicações web são fundamentais para proteger dados e usuários. Utilizando criptografia, e práticas de desenvolvimento seguro, é possível mitigar riscos, garantindo a integridade e confiabilidade do sistema. Estratégias como autenticação por token e hashing de senhas fortalecem a proteção, assegurando uma navegação online para o usuário segura e confiável (SOUZA et al., 2017).

- Passport-Local: De acordo com o site oficial do (LOCAL, 2014), este é uma estratégia de autenticação fornecida pelo Passport.js para autenticar usuários usando um nome de usuário e senha em aplicativos Node.js. Ele é facilmente integrado a qualquer aplicativo ou framework que suporte middlewares do estilo *Connect*, incluindo o Express. O Passport-local requer um retorno de chamada de verificação que valida as credenciais do usuário. Ele pode ser configurado para realizar a autenticação localmente, verificando o nome de usuário e a senha no banco de dados da aplicação.
- Token JWT: *JSON Web Token* (JWT) fornece uma abordagem segura para a troca de informações entre cliente e servidor por meio de um token gerado o qual tem como resultado final um objeto JSON, conforme explicado por Lucas, Ana e Jackson.

O token gerado pelo JWT é salvo no dispositivo do usuário e suas informações podem ser verificadas a cada solicitação, pois são criptografadas utilizando um segredo, através do algoritmo *HMAC* ou de um par de chaves públicas e privadas, garantindo assim a sua confiabilidade (SOUZA et al., 2017).

A Figura 3 exemplifica o fluxo de autenticação por token jwt:

Figura 3 – Fluxo de autenticação JWT



Fonte: [Souza et al. \(2017\)](#)

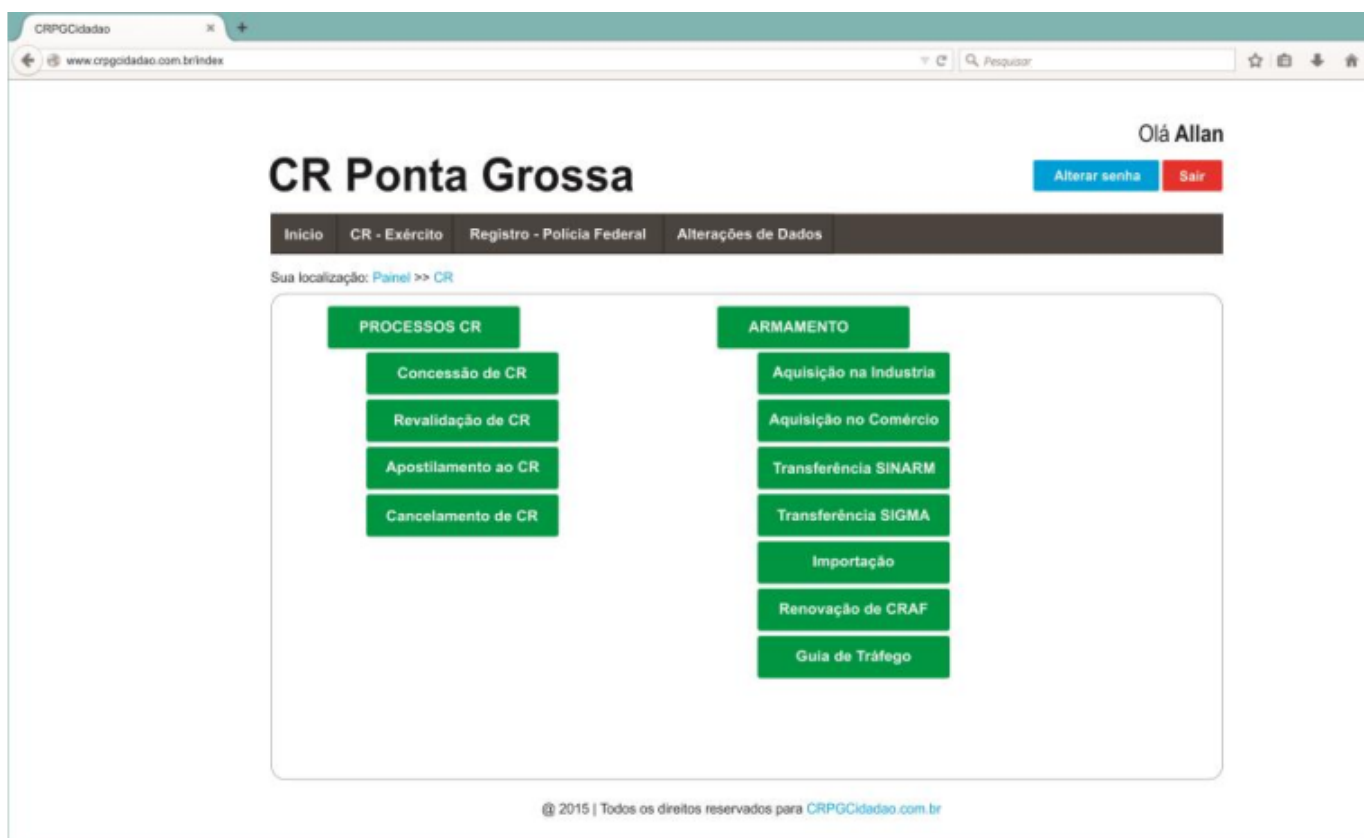
3 TRABALHOS RELACIONADOS

Nesta seção, são exploradas pesquisas que compartilham afinidades e características semelhantes com o sistema que este projeto busca desenvolver.

3.1 APLICAÇÃO WEB PARA PESSOAS FÍSICAS QUE UTILIZAM PRODUTOS CONTROLADOS PELO EXÉRCITO BRASILEIRO E POLÍCIA FEDERAL

A aplicação web denominada CRPG, tem o intuito de auxiliar os usuários a lidar com as etapas necessárias para a aquisição de produtos controlados pelo Exército Brasileiro e Polícia Federal. A [Figura 4](#) demonstra a tela de cadastro de processos.

Figura 4 – Tela de cadastros da aplicação CRPG



Fonte: [PAULA e Ramlow \(2015\)](#)

O Trabalho proposto se distingue da aplicação demonstrada acima, pois atua no âmbito de auxiliar os operadores da Brigada Militar afim de promover uma maior agilidade no processo de envio das informações para o SIGMA, buscando promover uma maior fluidez no controle e

regulamentação dos produtos controlados sob jurisdição do Exército que estão sob posse destes indivíduos e entidades mencionadas na [subseção 2.1.1](#).

3.2 SISTEMA DE GERENCIAMENTO DE LICENÇAS DE POSSE E PORTE DE ARMAS DE FOGO

Esta aplicação web se propõe a realizar todos os processos, como cadastro, validação de licenças, emissão de guias de tráfego, para indivíduos que tenham ou queiram adquirir o posse e porte de armas <<https://sartori-ria.github.io/tcc-gun-licence-control-spa/>>.

Figura 5 – Tela de cadastros da aplicação de Gerenciamento de Licenças

A imagem mostra a interface de usuário de uma aplicação web. À esquerda, há um menu lateral com as opções: 'Minhas Licenças', 'Nova Licença' (destacada), 'Agendamentos', 'Exames' e 'Atualizar meus dados'. O conteúdo principal, sob o título 'Nova Licença', apresenta duas colunas. A coluna da esquerda, intitulada 'Porte', lista os requisitos: 'Idade Mínima: 25' e 'Exames requeridos:' (com itens 'psicologico' e 'teste de tiro'). Abaixo, há um campo 'Endereço para a Licença' com uma seta para baixo e um botão 'Iniciar processo'. A coluna da direita, intitulada 'Posse', também lista os mesmos requisitos e possui um botão idêntico 'Iniciar processo'.

Fonte: [Sartori e Ramos \(2018\)](#)

O presente trabalho destaca-se da aplicação anterior ao concentrar seus esforços na assistência aos operadores da Brigada Militar, visando agilizar o envio de informações para o SIGMA. A intenção é favorecer uma maior fluidez no controle e regulamentação desses produtos controlados sob jurisdição do exército.

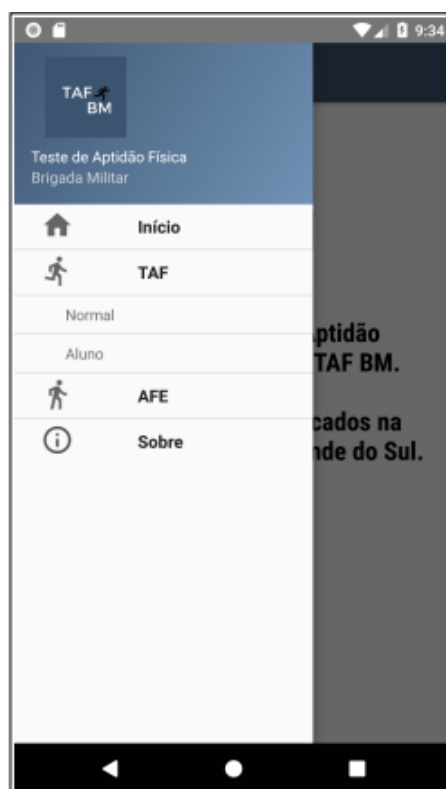
3.3 TAF- TESTE DE APTIDÃO FÍSICA DA BRIGADA MILITAR DO RIO GRANDE DO SUL

Um aplicativo móvel que visa facilitar a aplicação e avaliação do TAF na Brigada Militar do rio grande do sul. O TAF tem como objetivo verificar se os candidatos possuem as condições físicas mínimas exigidas para desempenhar as atividades do cargo ([MACHADO, 2018](#)).

O aplicativo foi desenvolvido em parceria do IFRS com a BM RS, proporcionando mais projetos como este para alunos da instituição, trazendo benefícios para órgãos governamentais, fortalecendo a integração prática do conhecimento acadêmico com as demandas da sociedade.

Na [Figura 6](#) é demonstrado a tela inicial do TAF

Figura 6 – Tela Inicial do aplicativo TAF BM



Fonte: [Machado \(2018\)](#)

Por se tratar de resoluções em contextos diferentes, o aplicativo “TAF” se encaixa nesta seção por ter sido desenvolvido para a Brigada Militar do Rio Grande do Sul.

4 METODOLOGIA

Neste segmento, serão delineadas informações relacionadas à metodologia utilizada para elaboração da aplicação proposta neste projeto. Sendo utilizada como base a metodologia de desenvolvimento de software iterativa e incremental. Este modelo enfatiza ciclos repetitivos curtos, chamados iterações, para construir o sistema incrementalmente. Cada iteração inclui planejamento, design, implementação e testes, permitindo feedback contínuo. Prioriza a adaptabilidade a mudanças, flexibilidade e entrega de funcionalidades incrementais, proporcionando entregas intermediárias e valor ao cliente ao longo do processo ([SOMMERVILLE, 2011](#)). Exemplos incluem metodologias ágeis como Scrum e Extreme Programming (XP).

Discutiremos também na estruturação do desenvolvimento, a análise de requisitos, abrangendo tanto os funcionais quanto os não funcionais.

Na delineação da Estruturação do desenvolvimento, é vital compreender a sinergia entre os componentes, visando otimizar a eficiência e a coesão do sistema em questão. Os requisitos funcionais, sendo pilares fundamentais, demandam uma abordagem analítica aprimorada para assegurar sua implementação fluída.

Quanto aos requisitos não funcionais, devemos explorar suas nuances de maneira minudente, reconhecendo a importância intrínseca de cada faceta no contexto global do projeto. Esta abordagem detalhada proporciona uma compreensão mais profunda, permitindo a delimitação precisa de parâmetros que transcendem as fronteiras meramente operacionais.

4.1 ESTRUTURAÇÃO DO DESENVOLVIMENTO

A forma como foi organizado o desenvolvimento do presente trabalho consiste inicialmente em reuniões realizadas através de entrevistas conduzidas com um membro da Brigada Militar, SD. Tiago Costa, por vezes juntamente com o orientador ou coorientador no *campus* Osório. Durante essas entrevistas, foram formuladas perguntas pertinentes ao procedimento de geração do AEL.

A coleta de informações foi complementada por meio de análise documental, sendo como principal embasamento a Portaria 136 de 8 de novembro de 2019 do Exército Brasileiro, mais especificamente o anexo D do [EXÉRCITO BRASILEIRO \(2019\)](#) constante nos Anexos (A, B, C, D). Esta documentação conceitua todas as fases envolvidas no processo de geração do AEL da Brigada Militar do Rio Grande do Sul.

A partir da análise de escopo realizada, planejou-se o uso do modelo de desenvolvimento iterativo neste projeto pois de acordo com [Sommerville \(2011\)](#), se um processo interno de desenvolvimento iterativo é usado, o documento de requisitos pode ser muito menos detalhado e

quaisquer ambiguidades podem ser resolvidas durante o desenvolvimento do sistema.

Logo a complexidade do procedimento de geração do AEL demanda uma abordagem flexível, permitindo ajustes contínuos à medida que novos pontos de vista são revelados durante entrevistas com SD. Tiago Costa. Essa metodologia proporciona validação incremental das fases, adaptabilidade a mudanças nos requisitos e envolvimento constante, garantindo um desenvolvimento mais preciso.

4.2 ANÁLISE DE REQUISITOS

Os requisitos inerentes a aplicação web, foram levantados com base nas informações coletadas durante as reuniões com os envolvidos conforme citado na [seção 4.1](#) sendo avaliado como as funcionalidades essenciais aquelas que impactam diretamente o pleno funcionamento da aplicação, quais devem ser implementadas durante o desenvolvimento do sistema. Ademais, foi realizada uma avaliação para determinar as funcionalidades desejáveis.

Sendo dividida esta em subseções contendo os requisitos funcionais e não funcionais.

4.2.1 Requisitos Funcionais

Nesta subseção, será abordado os requisitos funcionais, aqueles que abrangem e descrevem todas as funcionalidades previstas para o sistema.

Tabela 3 – Requisito Funcional 1

Identificador	Nome	Tipo	
RF-1	Cadastrar usuário operador	✓ Essencial	Desejável
Descrição			
O sistema deve permitir que usuários realizem o seu cadastro para obter acesso as funcionalidades da aplicação, Por padrão será atribuído perfil de operador e seu status definido como não autorizado, Deverá ser preenchido obrigatoriamente o campo ID. Funcionário(Identificação de funcionário), senha, e-mail e opcionalmente nome.			

Fonte: Autor

Tabela 4 – Requisito Funcional 2

Identificador	Nome	Tipo	
RNF-2	Armazenamento temporário	✓ Essencial	Desejável
Descrição			
O sistema deve suportar o armazenamento temporário de dados no Armazenamento local do navegador. Os registros temporários devem ser salvos neste armazenamento local.			

Fonte: Autor

Tabela 5 – Requisito Funcional 3

Identificador	Nome	Tipo	
RF-3	Autenticar usuários	✓ Essencial	Desejável
Descrição			
Deve permitir o acesso à aplicação para um usuário cadastrado através da identificação de funcionário e senha.			
Restrição			
Apenas usuários previamente autorizados pelo administrador podem ter acesso ao sistema.			

Fonte: Autor

Tabela 6 – Requisito Funcional 4

Identificador	Nome	Tipo	
RF-4	Acesso as configurações de usuários	✓ Essencial	Desejável
Descrição			
O sistema deve permitir o acesso ao menu de configuração para gerenciamento dos usuários, listando os usuários que realizaram o cadastro no sistema.			
Restrição			
Apenas usuários com perfil de administrador podem acessar essa opção do menu.			

Fonte: Autor

Tabela 7 – Requisito Funcional 5

Identificador	Nome	Tipo	
RF-5	Alterar senha de usuário	✓ Essencial	Desejável
Descrição			
Essa função deve permitir que um usuário administrador possa alterar a senha de outro usuário, apenas informando uma nova.			

Fonte: Autor

Tabela 8 – Requisito Funcional 6

Identificador	Nome	Tipo	
RF-6	Excluir usuário	✓ Essencial	Desejável
Descrição			
Essa função deve permitir que um usuário administrador possa excluir um usuário específico. O sistema deve solicitar a confirmação da ação a ser executada.			
Restrição			
O Usuário somente será excluído, mediante confirmação da ação a ser executada.			

Fonte: Autor

Tabela 9 – Requisito Funcional 7

Identificador	Nome	Tipo	
RF-7	Autorizar acesso de usuário	✓ Essencial	Desejável
Descrição			
Essa função deve permitir que um usuário administrador possa alterar o status de outro usuário para autorizado , assim permitindo o acesso ao sistema. O sistema deve solicitar a confirmação da ação a ser executada			
Restrição			
O Usuário somente será autorizado, mediante confirmação da ação a ser executada.			

Fonte: Autor

Tabela 10 – Requisito Funcional 8

Identificador	Nome	Tipo	
RF-8	Bloquear acesso do usuário	✓ Essencial	Desejável
Descrição			
Essa função deve alterar o status do usuário para não autorizado , assim bloqueando o acesso ao sistema. O sistema deve solicitar a confirmação da ação a ser executada.			
Restrição			
O Usuário somente será bloqueado, mediante confirmação da ação a ser executada.			

Fonte: Autor

Tabela 11 – Requisito Funcional 9

Identificador	Nome		
RF-9	Cadastrar arma	Essencial	✓ Desejável
Descrição			
O sistema deve permitir o cadastro de uma arma sendo necessário informar marca, espécie, modelo, calibre, grupo do calibre, tipo de funcionamento, quantidade de canos, comprimento do cano, unidade de medida do cano, tipo de alma, país e opcional, capacidade do cartucho, número de raia, sentido das raia e nome do acabamento.			

Fonte: Autor

Tabela 12 – Requisito Funcional 10

Identificador	Nome		
RF-10	Alterar arma	Essencial	✓ Desejável
Descrição			
O sistema deve permitir alterar as informações de uma arma cadastrada.			

Fonte: Autor

Tabela 13 – Requisito Funcional 11

Identificador	Nome		
RF-11	Salvar registro AEL temporariamente	✓ Essencial	Desejável
Descrição			
O sistema deve permitir o usuário salvar temporariamente um registro AEL informando obrigatoriamente: órgão, identificador utilizado pelo órgão, marca, espécie, modelo, calibre, grupo do calibre, tipo de funcionamento, quantidade de canos, comprimento do cano, unidade de medida do cano, tipo de alma, país, tipo de publicação, número do documento, data de publicação, CPF, nome, data de nascimento, número de identidade, data de expedição identidade, órgão emissor, UF do órgão emissor, nome do pai, nome da mãe, cidade residencial, tipo de proprietário e opcionalmente, capacidade do cartucho, número de raías, sentido das raías, nome do acabamento, órgão que publicou, profissão, logradouro comercial, bairro comercial, cidade comercial, logradouro residencial e bairro residencial.			

Fonte: Autor

Tabela 14 – Requisito Funcional 12

Identificador	Nome		
RF-12	Autopreencher AEL	Essencial	✓ Desejável
Descrição			
O sistema deve autopreencher os campos com os dados da arma, caso selecionada uma arma cadastrada pelo usuário.			

Fonte: Autor

Tabela 15 – Requisito Funcional 13

Identificador	Nome		
RF-13	Gerar AEL	✓ Essencial	Desejável
Descrição			
O sistema deve gerar automaticamente um arquivo de texto contendo no máximo 50 registros temporários.			
Cada registro temporário deve ocupar somente uma linha do arquivo de texto.			
Todos campos devem estar entre [] (colchetes), campos opcionais devem ser gerados com [] em branco e nada preenchido entre ele.			
Não poderá haver linha em branco no início do arquivo, entre registros ou após o último registro do arquivo.			
A primeira linha é reservada para inserir o cabeçalho do arquivo que obrigatoriamente deverá seguir o padrão [REMOTO][Data de Criação do Arquivo Hora de Criação do Arquivo][Número de Registros], e os registros devem ser inseridos da 2ª linha em diante.			

Fonte: Autor

Tabela 16 – Requisito Funcional 14

Identificador	Nome		
RF-14	Limpar registros temporários AEL	Essencial	✓ Desejável
Descrição			
O sistema deve permitir o usuário limpar os registros temporários. Para limpar os registros temporários deve ser solicitada a senha do usuário.			

Fonte: Autor

4.2.2 Requisitos Não Funcionais

Na presente subseção, serão apresentados os requisitos não funcionais da aplicação, os quais descrevem as características globais do sistema, indo além das funcionalidades específicas. E assegurando uma compreensão completa e abrangente dos métodos empregados pela aplicação para alcançar o resultado almejado.

Tabela 17 – Requisito Não Funcional 1

Identificador	Nome	Tipo	
RNF-1	Acesso à internet	✓ Essencial	Desejável
Descrição			
A aplicação web necessita de conexão a internet para manter seu pleno funcionamento.			

Fonte: Autor

Tabela 18 – Requisito Não Funcional 2

Identificador	Nome	Tipo	
RNF-2	Armazenamento temporário	✓ Essencial	Desejável
Descrição			
O sistema deve suportar o armazenamento temporário de dados no Armazenamento local do navegador. Os registros temporários devem ser salvos neste armazenamento local.			

Fonte: Autor

Tabela 19 – Requisito Não Funcional 3

Identificador	Nome	Tipo	
RNF -3	Segurança	✓ Essencial	Desejável
Descrição			
O sistema deve utilizar <i>JSON Web Tokens (JWT)</i> como mecanismo de autenticação. Os <i>tokens JWT</i> serão utilizados para garantir a segurança na comunicação entre o cliente e o servidor, fornecendo um método eficiente e seguro de verificar a identidade dos usuários.			

Fonte: Autor

Tabela 20 – Requisito Não Funcional 4

Identificador	Nome	Tipo	
RNF -4	Escalabilidade da infraestrutura	Essencial	✓ Desejável
Descrição			
O sistema deve ser projetado para ser escalável, capaz de lidar com um aumento significativo no número de usuários, tráfego e dados. A arquitetura da aplicação e dos servidores deve ser dimensionável para garantir o desempenho contínuo em cenários de carga variável.			

Fonte: Autor

Tabela 21 – Requisito Não Funcional 5

Identificador	Nome	Tipo	
RNF -5	Validação de erros para o usuário	✓ Essencial	Desejável
Descrição			
O sistema deve implementar uma robusta e amigável validação de erros, fornecendo retorno imediato e compreensível para o usuário em caso de entradas inválidas. Mensagens de erro devem ser claras, específicas e apresentadas de forma a orientar o usuário na correção do problema.			

Fonte: Autor

Tabela 22 – Requisito Não Funcional 6

Identificador	Nome	Tipo	
RNF -6	Usabilidade Intuitiva	✓ Essencial	Desejável
Descrição			
O sistema deve garantir uma <i>interface</i> de usuário intuitiva, facilitando a navegação e interação para usuários de diversos níveis de habilidade, promovendo uma experiência de uso eficaz e satisfatória.			

Fonte: Autor

Tabela 23 – Requisito Não Funcional 7

Identificador	Nome	Tipo	
RNF-7	Dados sensíveis	✓ Essencial	Desejável
Descrição			
O sistema deve garantir um tratamento adequado com os dados sensíveis que são inseridos na aplicação, deve ser utilizado uma abordagem para não armazenar essas informações particulares.			

Fonte: Autor

5 DESENVOLVIMENTO DA APLICAÇÃO AUTOFORM

Neste capítulo é apresentado o desenvolvimento do projeto, onde são apresentados, a arquitetura do sistema, o processo de produção, e a implementação do sistema.

5.1 PROCESSO DE PRODUÇÃO

O projeto teve sua fase inicial, após o 1º encontro com o cliente Tiago Costa e o coordenador Márcio Lemos no *campus* Osório do IFRS, na reunião foi demonstrada a necessidade pela BM RS de uma aplicação que pudesse ser utilizada para auxiliar no processo de registro do AEL no SIGMA, a fim de agilizar o processo de preenchimento e automatizar a geração do arquivo, bem como resolver outras dificuldades encontradas pelos operadores. Após a definição de contexto do AEL por parte da BM RS, foi realizado um estudo sobre o assunto, para que fosse possível entender o problema e propor uma solução viável e eficaz.

Com a elicitação dos requisitos concluída, ocorreu a prototipação das telas da aplicação, utilizando a ferramenta de *design de interface* de usuário [Figma \(2023\)](#), a fim de validar se o visual e o fluxo da aplicação iriam atender o propósito da instituição.

5.1.1 Tecnologias

Após a validação do protótipo, foi realizada a escolha das tecnologias que seriam utilizadas no desenvolvimento da aplicação, sendo elas o *framework* [Meta Open Source \(2013\)](#), para o desenvolvimento *frontend*, e o *framework* NodeJS [Node \(2009\)](#) para o desenvolvimento do *backend* da aplicação web, juntamente com o banco de dados não relacional [MongoDB \(2023\)](#), para o armazenamento dos dados.

Com as tecnologias definidas, foi realizado o desenvolvimento da aplicação web e hospedado em um servidor, para que fosse possível realizar os testes e validações com o cliente, e assim, realizar as correções necessárias.

Portanto após a primeira versão da aplicação estar disponível na *internet* as adaptações necessárias e atualizações eram organizadas e relatadas mediante troca de mensagens com o cliente.

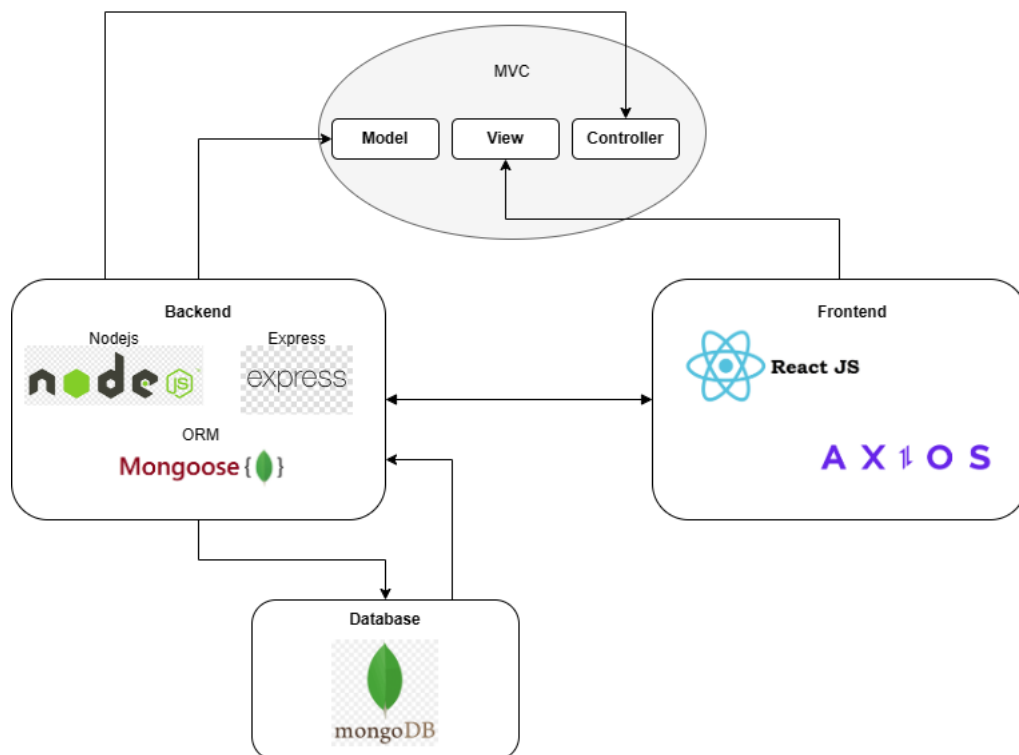
Para permitir que fosse possível realizar atualizações em produção na aplicação foi utilizada a ferramenta [Git \(2023\)](#), juntamente com a plataforma de hospedagem de código fonte [GitHub \(2023\)](#), para que fosse possível realizar o controle de versões e o versionamento do código fonte da aplicação, garantindo sempre uma versão estável e disponível em produção, enquanto era viável desenvolver paralelamente novas funcionalidades e validá-las com o cliente.

5.1.2 Arquitetura da aplicação

Nesta seção será abordado a visão de como foi projetado a estrutura do sistema que é composto por uma aplicação web, e uma API.

A Arquitetura *MVC* foi a escolhida para a estruturação do sistema, ficando então dispostas as tecnologias mencionadas na [subseção 5.1.1](#) da seguinte maneira ilustrada pela [Figura 7](#).

Figura 7 – Disposição das tecnologias na arquitetura MVC



Fonte: Autor

Toda comunicação entre a aplicação web e a API é realizada através de requisições HTTP, onde a aplicação web realiza requisições para a API, e a API responde com os dados solicitados, caso necessário a API busca as informações no banco de dados e retorna para a aplicação web.

As requisições e respostas seguem o padrão REST, onde cada requisição possui um método HTTP, e um *endpoint* que é a URL que identifica o recurso que está sendo solicitado. A resposta da requisição é um JSON, que contém os dados solicitados.

Atuando na camada de modelo da aplicação, o banco de dados é responsável por armazenar os dados da aplicação e foi construído utilizando dois modelos principais para armazenamento não estruturado, o primeiro modelo representa as informações referente as armas de acordo com a [Figura 8](#). Já o segundo modelo evidenciado na [Figura 9](#) é utilizado para representar as informações referente aos usuários da aplicação.

Figura 8 – Diagrama da estrutura de armas

```
Arma {  
  ID : INTEGER (PK, AI)  
  MarcaArma : STRING  
  EspécieArma : STRING  
  Modelo : STRING (15)  
  Calibre : STRING (30)  
  GrupoCalibreArma : STRING  
  CapacidadeCartucho : STRING  
  TipoFuncionamentoArma : STRING  
  QuantidadeCanos : STRING  
  ComprimentoCano : STRING  
  UnidadeMedidaCano : ENUM('CM', 'MM', 'POL')  
  TipoAlma : ENUM('L', 'R')  
  NumeroRaias : STRING  
  SentidoRaias : ENUM('E', 'D')  
  NomeAcabamento : STRING (30)  
  Pais : STRING  
}
```

Fonte: Autor

Figura 9 – Diagrama da estrutura de usuário

```
Usuário {  
  ID : INTEGER (PK, AI)  
  username : STRING  
  password : STRING  
  name : STRING  
  email : STRING (UNIQUE, REQUIRED)  
  isAdmin : BOOLEAN (DEFAULT: false)  
  isApproved : BOOLEAN (DEFAULT: false)  
}
```

Fonte: Autor

5.1.3 Aplicação Web AutoForm

5.1.4 Autenticação

A página para acessar a aplicação [Figura 10](#) e a página de registro [Figura 11](#) são de acesso público, portanto podem ser visualizadas sem a necessidade de autenticação, já para acessar as demais páginas é necessário realizar o login.

A estratégia empregada para o acesso consiste na autenticação por meio de token JWT. O usuário realiza o login, fornecendo sua identificação de funcionário e senha. Essas informações são então encaminhadas para a API por meio de uma requisição POST. A API valida no banco de dados a existência de um usuário com os dados fornecidos. Se a validação for bem-sucedida, a API gera um token de acesso e o retorna como resposta. Posteriormente, a aplicação decodifica

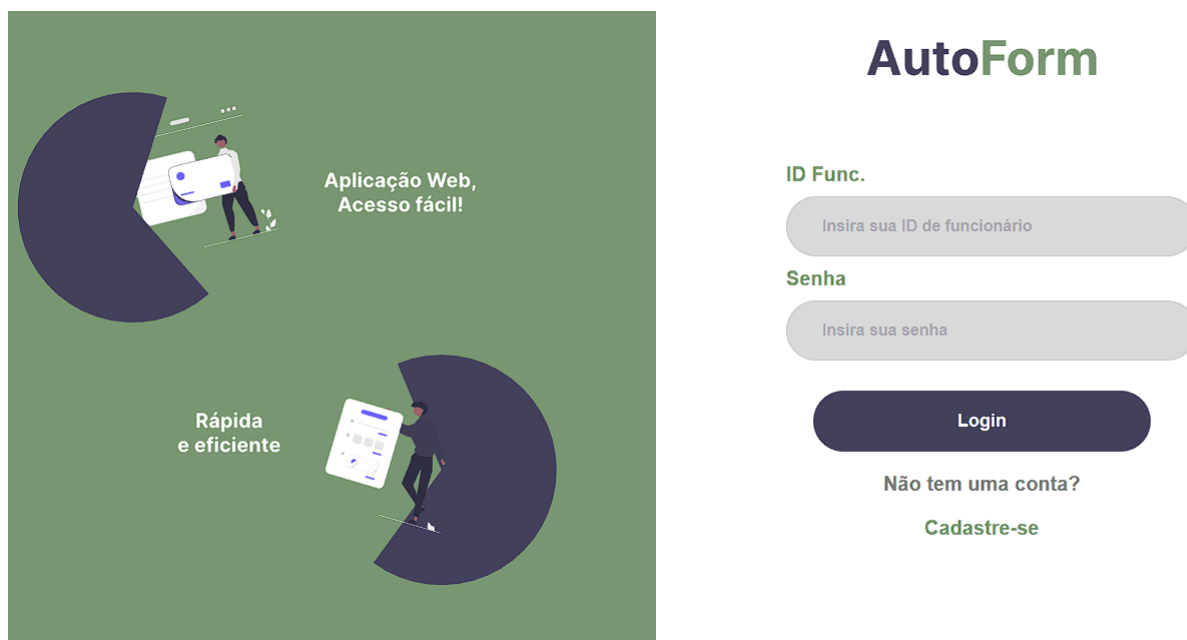
o token recebido, que contém as informações do usuário no *payload*, e armazena os dados de identificação no navegador da pessoa que solicitou o acesso.

A cada requisição posterior para outros *endpoints* que a aplicação realize para a API, o token é enviado no cabeçalho da requisição. A API irá validar somente o token, e caso seja válido retorna os dados solicitados.

5.1.5 Página de login

Esta página é responsável por realizar a autenticação do usuário, e para isso é necessário informar a identificação de funcionário e a senha. Caso o usuário não possua cadastro, é possível realizar o registro, clicando no botão "Registrar-se", que redireciona para a página de registro.

Figura 10 – Autoform - Página de login



Fonte: Autor

Se, por acaso, a validação no banco de dados não for bem-sucedida, a API emite uma resposta personalizada, indicando a natureza específica do impedimento.

Em um cenário em que a Identificação de Funcionário ou a senha fornecida não corresponde aos registros existentes, um alerta sutil é gerado, informando ao usuário sobre a disparidade. Este mecanismo proativo visa aprimorar a eficiência da correção do erro, direcionando o usuário a revisar e corrigir suas credenciais antes de prosseguir com a autenticação. Ademais, é importante ressaltar que, em situações extremas, como indisponibilidade temporária do serviço de autenticação, o usuário é informado sobre a impossibilidade de acesso, e orientado a tentar novamente mais tarde.

5.1.6 Página de registro

A página de registro [Figura 11](#) é responsável por realizar o cadastro de novos usuários, e para isso é necessário informar o email, a identificação de funcionário, a senha, e a confirmação da senha. E em caso de sucesso será retornada uma mensagem informando que o registro foi efetuado com sucesso.

Figura 11 – Autoform - Página de registros



← Ir para o início

AutoForm

Nome*

Email*

ID Func.*

Senha*

Confirme sua senha*

Cadastre-se

* Campos de preenchimento obrigatório

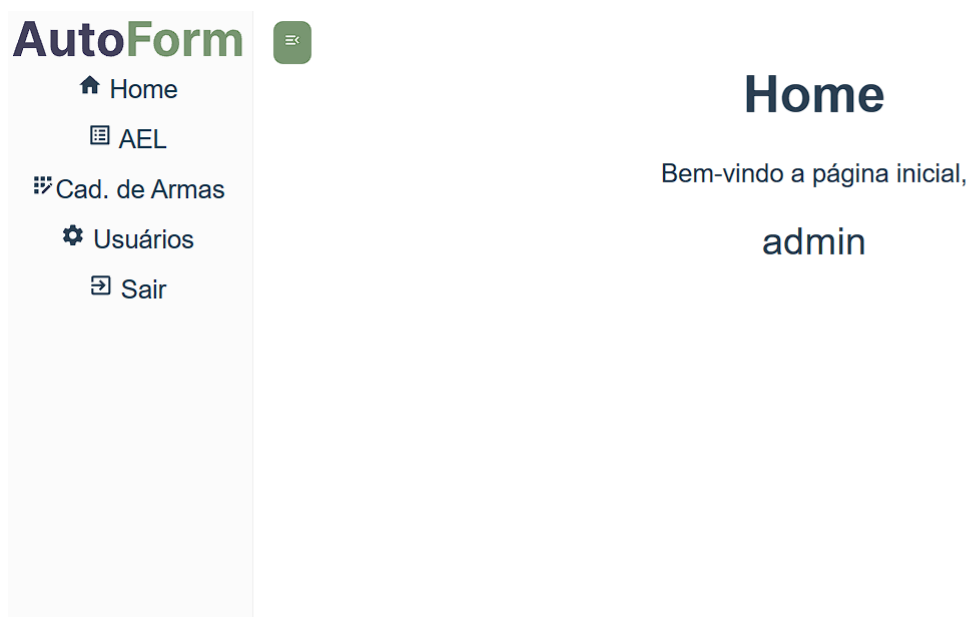
Fonte: Autor

Caso a identificação de funcionário já esteja cadastrada, será retornada uma mensagem informando que a identificação de funcionário já está cadastrada. Em caso de não conformidade da senha com a confirmação da senha, será retornada uma mensagem informando que as senhas não conferem.

5.1.7 Página inicial

Esta é a página inicial da aplicação, a qual o usuário é direcionado assim que realiza o login, e nela é possível visualizar no menu lateral as opções disponíveis de criação de AEL, cadastros, usuários e sair da aplicação.

Figura 12 – Autoform - Página inicial home



Fonte: Autor

5.1.8 Página de criação de AEL

A página de criação de AEL, [Figura 13](#), é responsável por realizar a criação de um registro e geração do AEL, sendo necessário informar os campos obrigatórios que constam no [Apêndice B](#) e [Apêndice C](#).

Cada registro é salvo no *local storage* do navegador do usuário com todas informações preenchidas no formulário, para posterior geração do arquivo.

Essa estratégia de armazenamento temporário foi adotada visando manter a integridade e a segurança das informações caso o usuário perca a conexão com a internet ou por algum outro motivo a aplicação do navegador seja encerrada, o que acarretaria na perda de todos os registros já digitados, utilizando essa abordagem é possível resgatar da memória do navegador todos os registros salvos temporariamente para geração do AEL, sem necessidade de armazenar no banco de dados ou enviar pela rede informações sensíveis como CPF entre outros.

5.1.9 Página de criação de AEL com opção de autopreenchimento

Na [Figura 14](#) podemos visualizar um input que exibe uma lista com as armas cadastradas, facilitando o preenchimento de diversos campos do formulário, pois permite ao usuário selecionar uma arma previamente cadastrada, e os respectivos campos do formulário serão preenchidos automaticamente.

Figura 13 – Autoform - Página de criação do AEL

AEL - ARQUIVO ELETRÔNICO EM LOTE

Orgão:

Informe o órgão

Identificador Orgão:

Informe o identificador do órgão

Dados da Arma*

Número de Série:

Número de Série

Escolha uma arma:

Escolha uma opção

Marcas e Modelos

Marca:

Marca

Espécie:

Espécie

Modelo:

Modelo

Calibre:

Calibre

Fonte: Autor

Figura 14 – Autoform - Página de criação do AEL com opção para selecionar

AEL - ARQUIVO ELETRÔNICO EM LOTE

☐ Home
 ☒ AEL
 ☐ Cad. de Armas
 ☐ Usuários
 ☐ Sair

Orgão:

Identificador Orgão:

Dados da Arma*

Número de Série:

Escolha uma arma:	
Marca: <input type="text" value="Marca"/>	<small>Escolha uma opção</small> Modelo G17 Marca 24 Calibre 9mm Modelo G22 Marca 1 Calibre 9MM Modelo T28 Marca 1 Calibre 9MM Modelo T78C Marca 1 Calibre 9MM Modelo PT192 Marca 1 Calibre 9MM Modelo C77R Marca 1 Calibre 9MM Modelo TH3 Marca 1 Calibre 9MM Modelo G3 TORO Marca 1 Calibre 9MM Modelo G3X1 Marca 1 Calibre 9MM Modelo G3C Marca 1 Calibre 9MM Modelo G18R Marca 24 Calibre 9MM Modelo AREX DELTA Marca 1121 Calibre 9MM Modelo CT140I Marca 1 Calibre .40 Modelo G3XL Marca 1 Calibre 9MM Modelo P7C3 Marca 1 Calibre 9MM Modelo P9 BRIGADE Marca 1 Calibre 9MM Modelo G3XR Marca 24 Calibre 9MM Modelo MASADAMARCA 1870 Calibre 9MM Modelo TP93 Marca 45 Calibre 9MM
Espécie: <input type="text" value="Espécie"/>	
Modelo: <input type="text" value="Modelo"/>	
Calibre: <input type="text" value="Calibre"/>	
Grupo do Calibre: <input type="text" value="Grupo do Calibre"/>	
Capacidade do Cartucho: <input type="text" value="Capacidade do Cartucho"/>	

Fonte: Autor

Figura 15 – Autoform - Página de criação do AEL -2

Tipo de Funcionamento:	<input type="text" value="Tipo de funcionamento"/>
Quantidade de Canos:	<input type="text" value="Quantidade de Canos"/>
Comprimento do Cano:	<input type="text" value="Comprimento do Cano"/>
Unidade Medida do Cano:	<input type="text" value="Escolha uma opção"/>
Tipo de Alma:	<input type="text" value="Escolha uma opção"/>
Número de Raias:	<input type="text" value="Número de Raias"/>
Sentido das Raias:	<input type="text" value="Escolha uma opção"/>
Nome do acabamento:	<input type="text" value="Nome do Acabamento"/>
País:	<input type="text" value="País"/>
Dados do Histórico*	
Tipo de Publicação:	<input type="text" value="Tipo de Publicação"/>
Número do Documento:	<input type="text" value="Número do Documento"/>
<input type="text"/>	

Fonte: Autor

Figura 16 – Autoform - Página de criação do AEL-3

Número do Documento:	<input type="text" value="Número do Documento"/>
Data de Publicação:	<input type="text" value="dd/mm/aaaa"/>
Orgão que Publicou:	<input type="text" value="Orgão que Publicou"/>
Dados do proprietário*	
CPF:	<input type="text" value="CPF"/>
Nome:	<input type="text" value="Nome"/>
Data de Nascimento:	<input type="text" value="dd/mm/aaaa"/>
Número da Identidade:	<input type="text" value="Número da Identidade"/>
Data Expedição ID:	<input type="text" value="dd/mm/aaaa"/>
Orgão Emissor:	<input type="text" value="Orgão Emissor"/>
UF Orgão Emissor:	<input type="text" value="UF Orgão Emissor"/>
Nome do Pai:	<input type="text" value="Nome do Pai"/>
Nome da Mãe:	<input type="text" value="Nome da Mãe"/>

Fonte: Autor

Figura 17 – Autoform - Página de criação do AEL-4

Nome do Pai:

Nome da Mãe:

Profissão:

Logradouro Comercial:

Bairro Comercial:

Cidade Comercial:

Logradouro Residencial:

Bairro Residencial:

Cidade Residencial:

Tipo Proprietário:

Seu arquivo já possui 2/50 linhas de dados

Fonte: Autor

Telas do formulário web para geração do AEL

Os botões da [Figura 17](#) tem as seguintes funcionalidades:

- Salvar: Verifica se todos os campos obrigatórios foram preenchidos, e caso estejam preenchidos, salva o registro no *local storage* do navegador.
- Gerar Arquivo: Com base nos registros salvos no *local storage*, a aplicação realiza a normalização das informações de acordo com as regras específicas de indexação contidas no [Apêndice A](#) e [Apêndice D](#).

Após uma caixa de pergunta é exibida ao usuário solicitando o nome do AEL e gera automaticamente o arquivo com base no nome inserido, realizando o download do arquivo AEL para o navegador do usuário que solicitou o recurso.

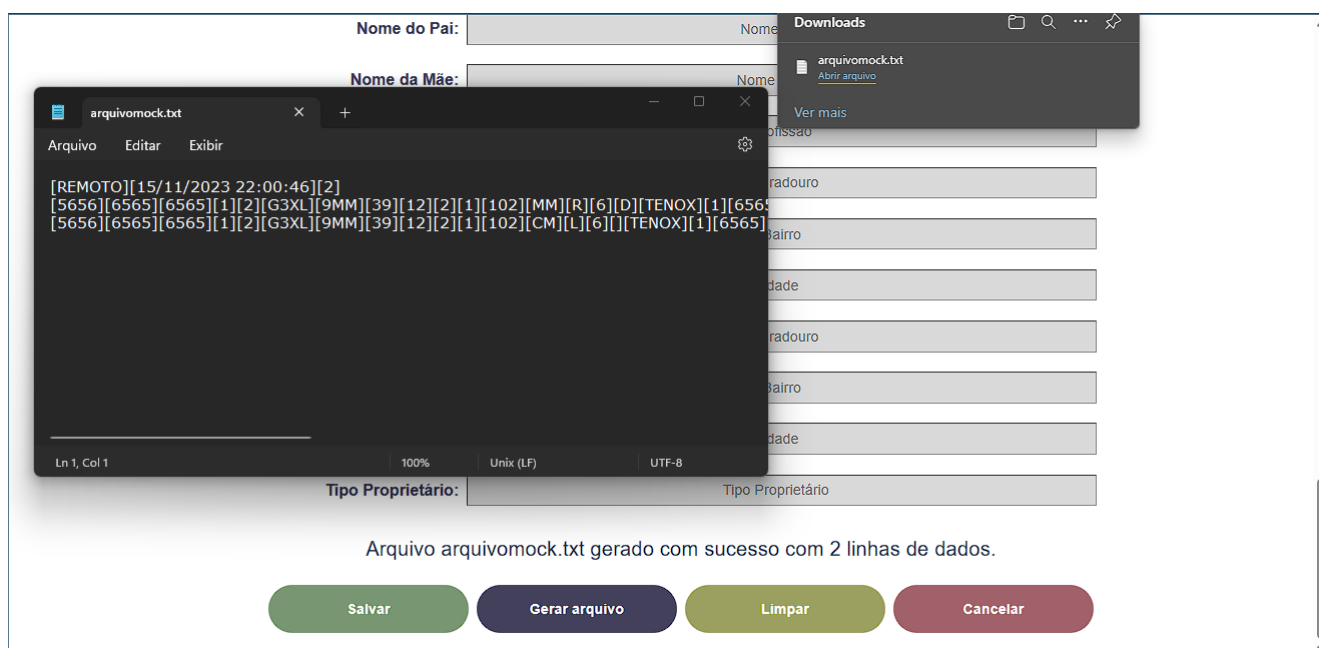
- Limpar: Este botão exibe uma mensagem perguntando ao usuário se deseja limpar N° linhas de registros contidos no armazenamento local do navegador. Após confirmada a ação os registros temporários são apagados. Esta opção foi implementada para garantir que após gerar o arquivo o usuário possa apagar os registros antigos para inserir novos. E também como garantia de segurança, limpando todos os dados do armazenamento temporário para que não fique nenhuma informação sensível salva no navegador do usuário.

- Cancelar: Esse botão tem como intuito apenas direcionar o usuário para a página inicial da aplicação a [Figura 12](#), saindo da opção de criação de AEL sem acarretar na perda dos registros temporários já salvos.

5.1.10 Exemplo de AEL gerado

Exemplo da ação executada após selecionar o botão gerar arquivo, que tem como saída o arquivo AEL formatado.

Figura 18 – Autoform - AEL gerado



Fonte: Autor

5.1.11 Página de cadastros

Nesta página é possível visualizar os cadastros de armas que são armazenadas no banco de dados, também é possível realizar uma pesquisa.

- Pesquisar: Neste campo é possível realizar uma pesquisa por uma arma específica, e caso seja encontrado o registro, o mesmo será exibido na tabela.
- Alterar: Neste botão é possível alterar o registro de arma. Após a edição o registro é atualizado no banco de dados.
- Excluir: Neste botão é possível excluir o registro de arma. Após a exclusão o registro é removido do banco de dados.

- **Cadastrar:** Neste botão é possível cadastrar uma nova arma, onde será exibido um modal contendo um formulário para o usuário preencher os dados da arma. Após a confirmação o registro é inserido no banco de dados e uma mensagem de sucesso é exibida ao usuário.

Figura 19 – Autoform - Cadastros



Bem-vindo a página de cadastros

Buscar Armas...

Especie	Modelo	Marca	Calibre	Ações
2	G17	24	9mm	Excluir Alterar
2	G2C	1	9MM	Excluir Alterar
2	TS9	1	9MM	Excluir Alterar
2	TH9C	1	9MM	Excluir Alterar
2	PT92	1	9MM	Excluir Alterar
4	CT9	1	9MM	Excluir Alterar
2	TH9	1	9MM	Excluir Alterar

Cadastrar

Fonte: Autor

5.1.12 Página de administração de usuários

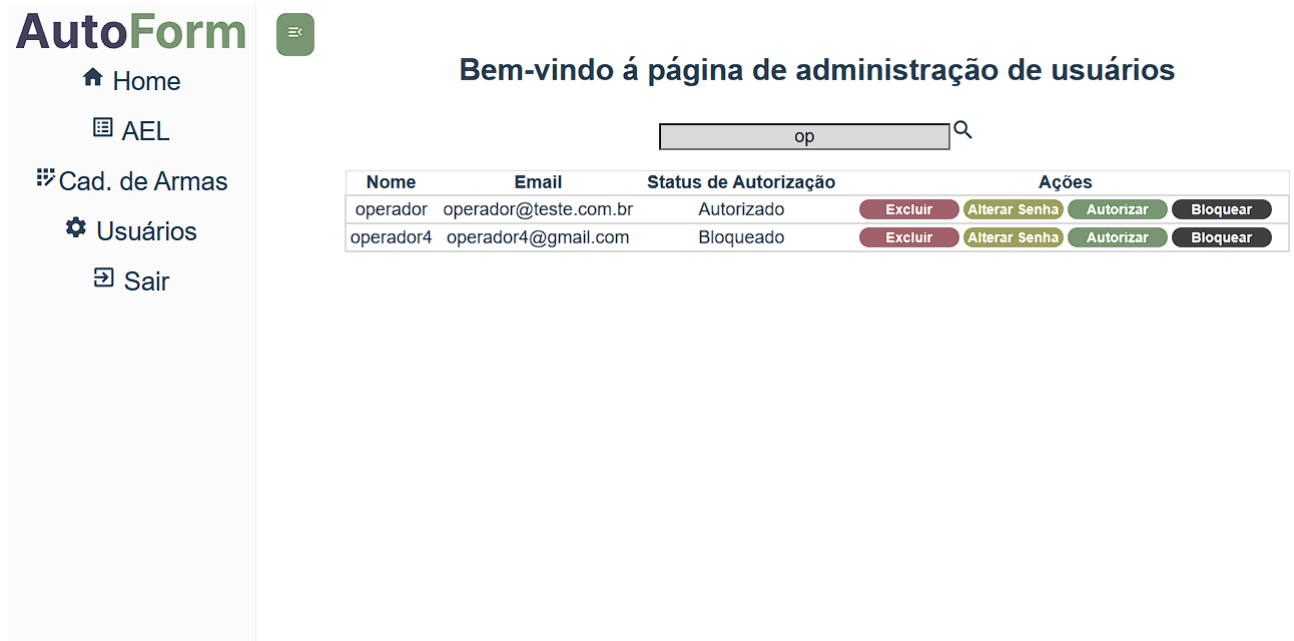
Esta página é acessível somente ao usuário com perfil de administrador. Nela é exibido o nome, e-mail e status de autorização dos usuários cadastrados.

Sendo possível realizar o gerenciamento destes através dos seguintes botões:

- **Pesquisar:** Neste campo é possível realizar uma pesquisa por um usuário específico. Caso seja encontrado o registro, o mesmo será exibido na tabela.
- **Excluir:** Neste botão é possível excluir o registro de usuário. Após a confirmação de exclusão o registro é removido do banco de dados.
- **Alterar senha:** Neste botão é possível alterar a senha de usuário, onde será exibido um campo para o administrador informar a nova senha e após a confirmação. O registro é atualizado no banco de dados e uma mensagem de sucesso é exibida ao usuário.
- **Autorizar:** Ao utilizar esse botão é possível alterar o status do usuário para autorizado, liberando o acesso a aplicação. O registro é atualizado no banco de dados e uma mensagem de sucesso é exibida ao usuário.

- Bloquear: Ao utilizar esse botão é possível alterar o status do usuário para bloqueado, bloqueando o acesso a aplicação. O registro é atualizado no banco de dados e uma mensagem de sucesso é exibida ao usuário.

Figura 20 – Autoform - Administração de usuários



AutoForm

- Home
- AEL
- Cad. de Armas
- Usuários
- Sair

Bem-vindo á página de administração de usuários

op

Nome	Email	Status de Autorização	Ações			
operador	operador@teste.com.br	Autorizado	Excluir	Alterar Senha	Autorizar	Bloquear
operador4	operador4@gmail.com	Bloqueado	Excluir	Alterar Senha	Autorizar	Bloquear

Fonte: Autor

6 CONCLUSÃO

O objetivo principal deste trabalho foi desenvolver uma solução que auxiliasse de forma eficaz os operadores da Brigada Militar ao realizar o processo de registro de produtos controlados no SIGMA através do AEL.

Para isso, foram levantadas as informações necessárias, através da elicitação de requisitos, pesquisa exploratória e análise documental para orientar a pesquisa em busca de uma solução eficiente que fosse segura, tivesse disponibilidade de acesso fácil, não armazenasse dados sensíveis, e ainda fosse capaz de minimizar o esforço empregado pelo operador para gerar o arquivo eletrônico.

Portanto o AutoForm atendeu aos objetivos específicos estabelecidos, proporcionando um preenchimento facilitado através de uma *interface* intuitiva e amigável, disponível de fácil acesso através da *web* e de forma segura, permitindo que diversos usuários realizem acesso simultâneo e ainda não armazenando dados sensíveis, pois o arquivo é gerado no computador do usuário.

A aplicação possibilita o cadastro de armas para todos os seus utilizadores, bem como o gerenciamento de usuários para o administrador viabilizando a supervisão por parte do administrador sobre quais pessoas têm acesso à aplicação.

6.0.1 Resultados finais

Vislumbrando-se uma estimativa de redução no tempo de preenchimento do formulário ao eleger uma arma previamente cadastrada para autopreencher 15 campos, seguida pela inserção de dados em apenas 24 campos remanescentes, contemplando inclusivamente os opcionais. Alternativamente, pode-se empregar o preenchimento de apenas 17 campos, sendo somente os obrigatórios, respectivamente, para a composição do arquivo eletrônico, outrora elaborado manualmente. Essa transição se baseia na totalidade de campos, quantificando em 39.

Logo esperasse que através deste trabalho tenha sido possível alavancar ainda mais a excelência operacional da Brigada Militar, contribuindo para a regulamentação dos produtos controlados e, consequentemente, para a segurança pública do estado do Rio Grande do Sul.

6.1 TRABALHOS FUTUROS

Nesta seção são descritos alguns trabalhos futuros que podem ser realizados para melhorar a aplicação.

Entre estas atualizações fica sugestão da implementação de alguns campos fixos no AEL como: tipo proprietário, órgão, órgão que publicou e profissão do proprietário, que não foram

implementados por não serem obrigatórios, mas que podem ser definidos futuramente por se tratar de informações fixas inerentes a instituição, logo irá minimizar ainda mais a quantidade de campos a serem preenchidos.

Na página de configurações seria interessante adicionar uma opção para o administrador poder selecionar algum outro usuário comum e inserir essa permissão especial, para que assim possa ser delegado a outra pessoa a responsabilidade de gerenciar os demais usuários. Também seria interessante a possibilidade de um botão com a opção para cadastrar novos usuários através desta página.

Na página do AEL, fica a sugestão para adicionar um modal que permita visualizar e editar os registros temporários já salvos, para que assim o usuário possa alterar algum dado que tenha sido preenchido incorretamente, ou até mesmo excluir o registro temporário.

REFERÊNCIAS

- BITENCOURT, A. S.; PAIVA, D. M. B.; CAGNIN, M. I. Elicitação de requisitos a partir de modelos de processos de negócio em bpmn: Uma revisão sistemática. *Anais Do XII Simpósio Brasileiro de Sistemas de Informação*, SBC, p. 200–207, 2016. Citado na página 18.
- BMRS. *Departamento Administrativo* — *bm.rs.gov.br*. 2023. <<https://www.bm.rs.gov.br/administrativo>>. [Accessed 07-11-2023]. Citado 2 vezes nas páginas 13 e 14.
- CARADA, F. *Entenda o que é Sinarm e Sigma - SHOT Assessoria e Registros* — *shot.net.br*. 2019. <<https://shot.net.br/entenda-o-que-e-sinarm-e-sigma/>>. [Accessed 11-11-2023]. Citado na página 16.
- CECHINEL, A. et al. Avaliação do framework angular e das bibliotecas react e knockout para o desenvolvimento do frontend de aplicações web. Florianópolis, SC, 2017. Citado na página 22.
- DÁVILA, G. A.; LEOCÁDIO, L.; VARVAKIS, G. Inovação e gerenciamento de processos: Uma análise baseada na gestão do conhecimento. *DataGramaZero-Revista de Ciência da Informação*, v. 9, n. 3, 2008. Citado na página 17.
- EXÉRCITO BRASILEIRO, M. D. D. *PORTARIA Nº136 - COLOG, DE 08 NOVEMBRO DE 2019*. 2019. <<http://www.dfpc.eb.mil.br/images/Portarian136.pdf>>. [Accessed 06-11-2023]. Citado 4 vezes nas páginas 13, 14, 17 e 29.
- FENEME. *Reunião sobre o sistema de gerenciamento militar de armas* — *feneme.org.br*. 2007. <<https://www.feneme.org.br/reuniao-sobre-o-sigma-sistema-de-gerenciamento-militar-de-armas/>>. [Accessed 11-11-2023]. Citado 2 vezes nas páginas 16 e 17.
- FIGMA. *Figma*. 2023. <<https://www.figma.com/>>. (Accessed on 11/15/2023). Citado na página 36.
- FIGUEIREDO, R. T. *Padrões de Projeto GOF aplicados ao Desenvolvimento de Jogos Eletrônicos*. 2014. <<https://repositorio.ufpe.br/handle/123456789/11981>>. (Accessed on 11/11/2023). Citado na página 18.
- FLANAGAN, D. *JavaScript: o guia definitivo*. [S.l.]: Bookman Editora, 2012. Citado 2 vezes nas páginas 19 e 21.
- GALHARDI, E. A. *Tema: Aplicações e Serviços WEB*. 2009. <<https://adelpha-api.mackenzie.br/server/api/core/bitstreams/01b5cabe-cd0c-4221-82a7-5e1521e3685b/content>>. (Accessed on 11/11/2023). Citado 2 vezes nas páginas 20 e 21.
- GIT. *Git*. 2023. <<https://git-scm.com/>>. (Accessed on 11/15/2023). Citado na página 36.
- GITHUB. *GitHub*. 2023. <<https://github.com/>>. (Accessed on 11/15/2023). Citado na página 36.
- LOCAL passport. *passport-local*. 2014. <<https://www.passportjs.org/packages/passport-local/>>. (Accessed on 11/12/2023). Citado na página 24.

- MACHADO, G. da S. *TAF - Teste de Aptidão Física da Brigada Militar do Rio Grande do Sul*. 2018. (Accessed on 11/12/2023). Citado 2 vezes nas páginas 27 e 28.
- MALDONADO, J. C. et al. Padrões e frameworks de software. *Notas Didáticas, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, ICMC/USP, São Paulo, SP, Brasil*, p. 28, 2002. Citado 2 vezes nas páginas 21 e 22.
- MDN. *JavaScript, MDN*. 2022. <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. [Accessed 11-11-2023]. Citado na página 21.
- Meta Open Source. *React - Meta Open Source*. 2013. <<https://opensource.fb.com/projects/react/>>. (Accessed on 11/12/2023). Citado 2 vezes nas páginas 23 e 36.
- MongoDB. *MongoDB*. 2023. <<https://www.mongodb.com/pt-br>>. (Accessed on 11/15/2023). Citado na página 36.
- NODE. *Node*. 2009. <<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>>. (Accessed on 11/12/2023). Citado 2 vezes nas páginas 22 e 36.
- OLIVEIRA, H. E. M. de. *Aplicativo Cliente-Servidor multicamadas para controle de uma rede de lojas via WEB utilizando Java*. 2003. <https://repositorio.ufsc.br/bitstream/handle/123456789/183897/tcc_Henrique_Oliveira.pdf?sequence=-1&isAllowed=y>. (Accessed on 11/11/2023). Citado na página 19.
- PAULA, J. P. F. D.; RAMLOW. *APLICATIVO WEB PARA PESSOAS FÍSICAS INTITULADAS ATIRADORES, COLECIONADORES, CAÇADORES E CIDADÃO COMUM QUE UTILIZAM PRODUTO CONTROLADO PELO EXÉRCITO BRASILEIRO E POLÍCIA FEDERAL*. 2015. <<https://educapes.capes.gov.br/handle/capes/661369?mode=full>>. (Accessed on 11/12/2023). Citado na página 26.
- PEREIRA, C. *web real-time com Node.js*. Casa do Código, 2014. ISBN 9788566250930. Disponível em: <<https://books.google.com.br/books?id=Wm-CCwAAQBAJ>>. Citado 3 vezes nas páginas 22, 23 e 24.
- REDES. 2016. <https://www.gta.ufrj.br/ensino/eel878/redes1-2016-1/16_1/p2p/modelo.html>. (Accessed on 11/11/2023). Citado na página 19.
- SARTORI, L. A.; RAMOS. *Sistema de gerenciamento de licenças de posse e porte de armas de fogo*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2018. Citado na página 27.
- SOMMERVILLE, I. *Engenharia de software*. Pearson Prentice Hall, 2011. (Accessed on 11/12/2023). ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>. Citado na página 29.
- SOUZA, L. et al. *Utilização de JSON Web Token na Autenticação de Usuários em APIs REST*. 2017. <https://www.enacompt.com.br/2017/docs/json_web_token_api_rest.pdf>. (Accessed on 11/12/2023). Citado 2 vezes nas páginas 24 e 25.
- VALENTE, M. T. *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. [S.l.]: Editora: Independente, 2020. Citado na página 20.

ANEXO A – PORTARIA 136 - ANEXO D.1

Anexo D

CADASTRO DE ARMA DE FOGO NO SIGMA VIA ARQUIVO ELETRÔNICO EM LOTE (AEL)

1. FINALIDADE

O cadastro de armas arquivo eletrônico em lote permite que o procedimento seja simplificado e mantenha o controle dos dados, a fim de obter celeridade nos processos de registro de arma no SIGMA.

2. OBJETIVO

O cadastro de armas de fogo no SIGMA requer publicação em documento oficial permanente do órgão de vinculação do adquirente, conforme o art. 3º da Lei 10826, de 22 de novembro de 2003 (Estatuto do Desarmamento). A publicação deve conter as informações previstas no art. 5º do Decreto nº9847, de 25 de junho de 2019. O cadastro no SIGMA, via arquivo eletrônico em lote, visa a formação do número de série da arma, a inserção dos dados e a habilitação para a emissão do Certificado de Registro de Arma de Fogo.

3. FASES DO PROCEDIMENTO

3.1. Publicação em documento oficial permanente do órgão de vinculação do adquirente

O registro das informações de armas de fogo deve constar de documentos oficiais permanentes da instituição para posterior cadastro no Sistema de Gerenciamento Militar de Armas (SIGMA).

3.2. Preenchimento do arquivo eletrônico

3.2.1. Os arquivos eletrônicos em lote (AEL) são no formato texto e devem atender a um layout pré-definido.

3.2.2. O arquivo tem o formato texto (TextEncoding = ISO-8859-1), com no máximo 10 MB (10240 Kbytes) de tamanho.

3.2.3. O nome do arquivo deverá obrigatoriamente ser gerado pelo operador, devendo seguir as orientações:

a) O nome do arquivo deverá obrigatoriamente ser composto por:

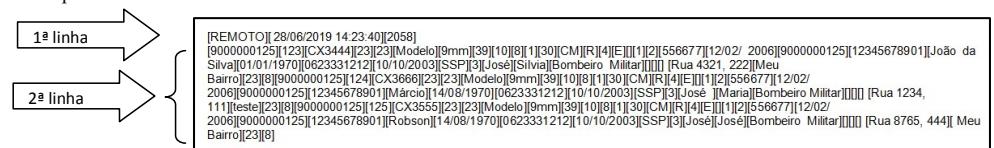
“CARGA-“ código do órgão “-“ data de geração “-“ hora da geração “.txt”

b) O código do órgão será gerado ao inserir seus dados no sistema SIGMA. Para isso, é importante que os órgãos que contiverem erros em seus dados, deverão informar à DFPC pelo e-mail cargasigma@dfpc.eb.mil.br. Os novos códigos gerados serão remetidos posteriormente, via e-mail encaminhado ao GSI, ABIN e às Polícias Militares e Bombeiros Militares.

c) Exemplo de nome de arquivo: supondo que o código do órgão gerador seja “9000000125” e a data e hora de geração sejam respectivamente “28/06/2019(dd/mm/aaaa)” e “14:23:40(hh:mm:ss)”. Dessa forma, o nome de arquivo será:

CARGA-900000125-28062019-142340.txt

3.2.4. Exemplo do AEL



3.2.5. Preenchimento da 1ª Linha do AEL

a) A primeira linha (cabeçalho) do arquivo obrigatoriamente deverá conter:

[REMOTO][Data de Criação do Arquivo Hora de Criação do Arquivo][Número de Registros]

b) Detalhamento dos campos do cabeçalho:

[REMOTO] – Informação de controle. Deverá conter a palavra “REMOTO” em letras maiúsculas.

[Data de Criação do Arquivo] – Data que o arquivo foi gerado. Deverá estar no formato dd/mm/aaaa.

[Hora de Criação do Arquivo] – Hora que o arquivo foi gerado. Deverá estar no formato hh:mm:ss.

[Número de Registros] – Conterá o número de linha/registros que contém o arquivo, excluindo a primeira linha (cabeçalho) do arquivo.

c) Exemplo da primeira linha do arquivo: supondo que a data e hora de geração sejam respectivamente

“28/06/2019(dd/mm/aaaa)” e “14:23:40(hh:mm:ss)”, e que o arquivo contenha 2058 linhas, excluindo o cabeçalho. A primeira linha será:

[REMOTO][28/06/2019 14:23:40][2058]

ANEXO B – PORTARIA 136 - ANEXO D.2

3.2.6. Preenchimento da 2ª Linha do AEL

a) Na segunda linha do arquivo, cada registro/linha deverá ser composto pelos dados abaixo, em uma única linha, obrigatoriamente na ordem em que aparecem e sempre entre colchetes.

[Órgão][Identificador Utilizado pelo Órgão][Número de Série][Marca da Arma][Espécie da Arma][Modelo][Calibre][Grupo do Calibre][Capacidade do Cartucho][Tipo de Funcionamento][Quantidade de Canos][Comprimento do Cano][Unidade de Medida do Cano][Tipo de Alma][Número de Raias][Sentido das Raias][Nome do Acabamento][País][Tipo de Publicação][Número do Documento de Ocorrência][Data de Publicação][Órgão que Publicou][CPF] [Nome][Data de Nascimento][Número Identidade][Data de Expedição Identidade][Órgão Emissor][UF do Órgão Emissor][Nome do Pai][Nome da Mãe][Profissão][Logradouro Comercial][Bairro Comercial][Cidade Comercial][Logradouro Residencial][Bairro Residencial][Cidade Residencial][Tipo de Proprietário da Arma]

b) Os campos de um registro/linha do arquivo estão detalhados na TABELA DE DETALHAMENTO DOS CAMPOS que segue:

POSIÇÃO DO CAMPO	NOME DO CAMPO	OBRIGATÓRIO	TIPO DO CAMPO	DESCRIÇÃO
Informações Gerais (obrigatórias para todas as linhas do arquivo)				
1	[Órgão]	S	Numérico	Código dos órgãos que enviou o arquivo ao Exército. Obtido na tabela ORGAO. Posteriormente disponível no site e encaminhado via ofício.
2	[Identificador Utilizado pelo Órgão]	S	Numérico	Identificador único da arma no órgão.
Dados da Arma				
3	[Número de Série]	S	Texto (20)	Número de identificação existente na arma.
4	[Marca da Arma]	S	Numérico	Código obtido da tabela MARCA_ARMA. Disponível no site. Lista de marcas de fabricante da arma.
5	[Espécie da Arma]	S	Numérico	Código obtido da tabela ESPECIE_ARMA, disponível no site. Lista de espécie das armas registradas.
6	[Modelo]	S	Texto (15)	Nome dado pelo fabricante para uma determinada arma.
7	[Calibre]	S	Texto (30)	Descrição do calibre da arma conforme especificado pelo fabricante.
8	[Grupo do Calibre]	S	Numérico	Código do grupo de calibres obtido da tabela GRUPO_CALIBRE_ARMA.
9	[Capacidade do Cartucho]	N	Numérico (3)	Quantidade máxima de cartuchos ou tiros que a arma pode suportar em suas câmaras, tambor ou carregador.
10	[Tipo de Funcionamento]	S	Numérico	Código do tipo de funcionamento obtido da tabela TIPO_FUNCIONAMENTO_ARMA.
11	[Quantidade de Canos]	S	Numérico (2)	Número de canos existentes na arma.
Dados da Arma				
12	[Comprimento do Cano]	S	Numérico (3,2)	Número da medida de comprimento do cano.
13	[Unidade de Medida do Cano]	S	Texto (3)	Unidade de medida do comprimento do cano. Opções de preenchimento: “CM” para centímetro, “MM” para milímetro, “POL” para polegada.
14	[Tipo de Alma]	S	Texto (1)	Tipo de alma do cano. Opções de preenchimento: “L” para alma lisa, “R” para alma raiada.
15	[Número de Raias]	N	Numérico (2)	Quantidade de raias do cano.
16	[Sentido das Raias]	N	Texto (1)	Sentido da raia do cano. “E” para a esquerda, “D” para a direita.
17	[Nome do Acabamento]	N	Texto (30)	Tipo do acabamento externo aplicado na arma.
18	[País]	S	Numérico	Código do país obtido da tabela PAIS do SIGMA.

ANEXO C – PORTARIA 136 - ANEXO D.3

Dados do Histórico (documento de publicação da arma)				
19	[Tipo de Publicação]	S	Numérico	Código do tipo de publicação obtido da tabela TIPO_PUBLICACAO_PRODUTO_CTRLDO do SIGMA
20	[Número do Documento]	S	Numérico (11)	Número do documento em que foi publicada a ocorrência.
21	[Data de Publicação]	S	Data	Data do documento em que foi publicada a ocorrência, no formato DD/MM/YYYY.
22	[Órgão que Publicou]	N	Numérico	Código do órgão que publicou a ocorrência. O código do órgão deve ser obtido da tabela ORGAO do SIGMA.
Dados do Proprietário da Arma				
23	[CPF]	S	Numérico (11)	Número do CPF da pessoa física. Identificador único do proprietário. O CPF deverá estar no formato 99999999999, sem “.” nem “-”
24	[Nome]	S	Texto (50)	Nome completo do proprietário
25	[Data de Nascimento]	S	Data	Data de nascimento no formato DD/MM/YYYY.
26	[Número Identidade]	S	Texto (20)	Número do documento de identidade.
27	[Data de Expedição Identidade]	S	Data	Data de expedição do documento de identificação no formato DD/MM/YYYY.
28	[Órgão Emissor]	S	Texto (30)	Nome do órgão que emitiu o documento de identificação.
29	[UF do Órgão Emissor]	S	Numérico	Código da Unidade Federal obtido da tabela UF. Disponível no site. UF do órgão que emitiu o documento de identificação.
30	[Nome do Pai]	S	Texto (50)	Nome do pai.
31	[Nome da Mãe]	S	Texto (50)	Nome da mãe.
32	[Profissão]	N	Texto (240)	Nome da Profissão. Ex: “Policial Militar”; “Bombeiro Militar”; Integrante da ABIN”; “Integrante do GSI”
33	[Logradouro Comercial]	N	Texto (60)	Descrição do endereço (Rua, Av., Rod, Nr, complemento) do local de trabalho.
34	[Bairro Comercial]	N	Texto (40)	Nome do bairro do local de trabalho.
35	[Cidade Comercial]	N	Numérico	Código da cidade obtido da tabela CIDADE do SIGMA
36	[Logradouro Residencial]	N	Texto (60)	Descrição do endereço (Rua, Av, Rod, Nr, complemento) de residência.
37	[Bairro Residencial]	N	Texto (40)	Nome do bairro onde reside.
38	[Cidade Residencial]	S	Numérico	Código da cidade obtido da tabela CIDADE do SIGMA
39	[Tipo de Proprietário]	S	Numérico	Código do tipo de proprietário da arma, obtido da tabela TIPO_PROPRIETARIO_ARMA do SIGMA.

c) Exemplo de um registro em um arquivo (com apenas uma única linha do arquivo):

[9000000125][123][CX3444][23][23][Modelo][9mm][39][10][8][1][30][CM][R][4][E][1][2][556677][12/02/2006][9000000125][12345678901][João][14/08/1970][0623331212][10/10/2003][SSP][3][José][Maria][Bombeiro Militar][1][1][Rua 1234, 111][teste][23][8]

d) Exemplo de um arquivo completo, contendo três registros:

[REMOTO][28/06/2019 14:23:40][2058]
[9000000125][123][CX3444][23][23][Modelo][9mm][39][10][8][1][30][CM][R][4][E][1][2][556677][12/02/2006][9000000125][12345678901][João da Silva][01/01/1970][0623331212][10/10/2003][SSP][3][José][Silvia][Bombeiro Militar][1][1][Rua 4321, 222][Meu Bairro][23][8]

ANEXO D – PORTARIA 136 - ANEXO D.4

[9000000125][124][CX3666][23][23][Modelo][9mm][39][10][8][1][30][CM][R][4][E][1][2][556677][12/02/2006][9000000125][12345678901][Márcio][14/08/1970][0623331212][10/10/2003][SSP][3][José][Maria][Bombeiro Militar][1][1][Rua 1234, 111][teste][23][8]
[9000000125][125][CX3555][23][23][Modelo][9mm][39][10][8][1][30][CM][R][4][E][1][2][556677][12/02/2006][9000000125][12345678901][Robson][14/08/1970][0623331212][10/10/2003][SSP][3][José][José][Bombeiro Militar][1][1][Rua 8765, 444][Meu Bairro][23][8]

e) Não poderá haver linha em branco no início do arquivo, entre registros ou após o último registro do arquivo.

f) Dados de preenchimento opcional, deverão obrigatoriamente conter os colchetes “[]”, e nada preenchido entre eles quando não contiverem dados.

g) Exemplo de parte de um registro/linha com preenchimento opcional:

... [Calibre][Grupo Calibre][Capacidade do Cartucho][Tipo de Funcionamento] ...

Na TABELA DE DETALHAMENTO DOS CAMPOS, [Capacidade do Cartucho] é um dado opcional. Então, caso não seja preenchido, o registro seria:

... [9mm][39][1][2] ...

Os valores [39] e [2] do exemplo, referem-se respectivamente aos códigos obtidos da TABELA DE DETALHAMENTO DOS CAMPOS.

3.3 Envio do arquivo à DFPC

3.3.1. O envio do AEL à Diretoria de Fiscalização de Produtos Controlados deve ser feito por meio eletrônico funcional da instituição para cargasigma@dfpc.eb.mil.br

3.4. Resposta da DFPC

A resposta da DFPC será também por meio eletrônico (Arquivo Resposta) o qual terá o seguinte conteúdo:

a) 1ª linha do Arquivo Resposta

[REMOTO][Data de Criação do Arquivo Hora de Criação do Arquivo][RESPOSTA]

b) 2ª linha do Arquivo Resposta

[SITUAÇÃO][Código do órgão][Nr série][Nr SIGMA]

c) Exemplo de Arquivo Resposta

[REMOTO][19/06/2019 13:03:59][4][RESPOSTA]

[OK][900000422][22275][1035724]

[OK][900000422][22277][1035725]

[OK][900000422][22280][1035726]

[OK][900000422][22281][1035727]

Nesse caso o AEL não apresentou erros no seu processamento e o SIGMA atribuiu o [Nr SIGMA] para 4 armas da instituição.

4. EMISSÃO DE CRAF

De posse do Arquivo Resposta da DFPC, o órgão de vinculação do interessado está habilitado a emitir o CRAF da arma cadastrada no SIGMA.

5. CONTATO TÉCNICO

O contato técnico das instituições com a DFPC deve ser feito por meio eletrônico funcional da instituição para cargasigma@dfpc.eb.mil.br