Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Arquitetura para Integração de Dados de Simulação de Combate no Âmbito da Força Terrestre

José Niuton da Nova

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Aletéia Patrícia Favacho de Araújo von Paumgartten

Coorientador
Prof. Dr. Flávio de Barros Vidal

Brasília
2025

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

da Nova, José Niuton
dN936a Arquitetura para Integração de Dados de Simulação de Combate no Âmbito da Força Terrestre / José Niuton da Nova; orientador Aletéia Patrícia Favacho de Araújo von Paumgartten; co-orientador Flávio de Barros Vidal. Brasília, 2025.
98 p.

Dissertação(Mestrado Profissional em Computação Aplicada) Universidade de Brasília, 2025.

1. Simulação Distribuída. 2. Simulação de Combate. 3. Integração de Simuladores. 4. Integração de Dados. I. Favacho de Araújo von Paumgartten, Aletéia Patrícia, orient. II. de Barros Vidal, Flávio, co-orient. III. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Arquitetura para Integração de Dados de Simulação de Combate no Âmbito da Força Terrestre

José Niuton da Nova

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Aletéia Patrícia Favacho de Araújo von Paumgartten (Orientador)
CIC/UnB

Prof. Dr. Edison Ishikawa
CIC/UnB

Prof. Dr. Maria Clicia Stelling de Castro
IME/UERJ

Prof. Dr. Edna Dias Canedo
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 24 de Outubro de 2025

Dedicatória

Dedico este trabalho à minha esposa, *Michelle*, e às minhas filhas amadas, *Maria Antônia* e *Maria Valentina*. Vocês são a razão da minha perseverança e a inspiração que me conduz. Que cada página aqui escrita seja também uma pequena homenagem ao amor, à paciência e ao apoio incondicional que sempre recebi de vocês.

Agradecimentos

Agradeço ao *Exército Brasileiro*, na pessoa do atual *Chefe do Preparo da Força Terrestre*, *Gen Bda Alexandre Pfaender Júnior*, e de seus ilustres antecessores, os quais compreenderam a relevância desta proposta de pesquisa.

Sou grato, também, aos militares da *Chefia do Preparo da Força Terrestre*, em especial aos integrantes da *Divisão de Sistemas e Subprograma/Sistema de Simulação da Força Terrestre*, cujo suporte moral e pessoal foi fundamental para a realização deste mestrado.

Registro, por fim, minha profunda gratidão aos meus orientadores, pela generosidade, paciência e incansável dedicação em compartilhar conhecimento e apontar caminhos, permitindo que este trabalho se concretizasse.

Resumo

No âmbito do avanço tecnológico em defesa, com ênfase em tecnologias disruptivas como *Big Data*, *Analytics* e Inteligência Artificial (IA), o Exército Brasileiro (EB) busca integrar essas ferramentas ao preparo da Força Terrestre (F Ter). Sistemas de simulação de combate produzem volumes elevados de dados, porém, a ausência de uma infraestrutura unificada para coleta, armazenamento e processamento impede a extração de valor completo desses dados, inviabilizando análises avançadas e aplicações de IA no treinamento e na experimentação doutrinária. O presente trabalho tem por finalidade propor uma arquitetura para integração de dados de simulação de combate, capaz de coletar, armazenar, processar e servir tais dados, atendendo às necessidades do Preparo da Força Terrestre e da Doutrina Militar Terrestre. Estudos sobre conceitos fundamentais em modelagem e simulação militar, engenharia de dados, arquiteturas de referência e *frameworks* de simulação fundamentaram a proposta, alinhada à visão de negócio. Um protótipo foi implementado e validado por meio de um estudo de caso, utilizando a metodologia *Goal-Question-Metric (GQM)*, que aferiu métricas de qualidade como eficiência (acima de 95% em componentes chave), confiabilidade (taxa de perda de 1%) e escalabilidade. Os testes revelaram aderência às características de *Big Data*, com geração de até 3 milhões de mensagens em 3 horas, baixa latência (0,1891 ms) e robustez no processamento de volumes elevados. Em conclusão, a arquitetura proposta demonstra viabilidade técnica e valor estratégico para o EB, superando limitações atuais e pavimentando o caminho para aplicações em *analytics* e IA, com recomendações para testes em maior escala na EBNet.

Palavras-chave: Simulação Distribuída, Simulação de Combate, Integração de Simuladores, Integração de Dados

Abstract

Within the scope of technological advancements in defense, with an emphasis on disruptive technologies such as Big Data, Analytics, and Artificial Intelligence, the Brazilian Army seeks to integrate these tools into the preparation of the Land Force. Combat simulation systems generate large volumes of data, however, the lack of a unified infrastructure for collection, storage, and processing prevents the complete extraction of value from this data, making advanced analytics and Artificial Intelligence applications in training and doctrinal experimentation unfeasible. This work aims to propose an architecture for the integration of combat simulation data, capable of collecting, storing, processing, and serving such data, meeting the needs of Land Force Preparation and Land Military Doctrine. Studies on fundamental concepts in military modeling and simulation, data engineering, reference architectures, and frameworks for simulation underpinned the proposal, aligned with the business vision. A prototype was implemented and validated through a case study, using the Goal-Question-Metric (GQM) methodology, which assessed quality metrics such as efficiency (above 95% in key components), reliability (1% loss rate), and scalability. Tests revealed compliance with Big Data characteristics, with the generation of up to 3 million messages in 3 hours, low latency (0.1891 ms), and robustness in processing large volumes. In conclusion, the proposed architecture demonstrates technical feasibility and strategic value for the Brazilian Army, overcoming current limitations and paving the way for applications in analytics and Artificial Intelligence, with recommendations for larger-scale tests on the Army Intranet.

Keywords: Distributed Simulation, Defense Simulation, Simulation Integration, Data Integration

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AED Ação Estratégica de Defesa.

AFSIM *Advanced Framework for Simulation, Integration and Modeling.*

AMAN Academia Militar das Agulhas Negras.

ANAC Agência Nacional de Aviação Civil.

API *Application Programing Interface.*

AQ Atributos de Qualidade.

ASA Ambiente de Simulação Aeroespacial.

AvEx Aviação do Exército.

BD *Big Data.*

BDA *Big Data Analytics.*

C Mil A Comando Militar de Área.

C4ISR *Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaisance.*

CA Centros de Adestramento.

CA-LESTE Centro de Adestramento - Leste.

CA-SUL Centro de Adestramento - Sul.

CDS Centro de Desenvolvimento de Sistemas.

CEE Concepção Estratégica do Exército.

CI Centros de Instrução.

CI Art Msl F Centro de Instrução de Artilharia de Mísseis e Foguetes.

CI Bld Centro de Instrução de Blindados.

CIAvEx Centro de Instrução de Aviação do Exército.

CID Centros de Integração de Dados.

COTER Comando de Operações Terrestres.

COTS *Commercial Off-The-Shelf*.

CPU *Central Processing Unit*.

CRFB Constituição da República Federativa do Brasil de 1988.

DAG *Directed Acyclic Graph*.

DIS *Distributed Interactive Simulation*.

DoD Department of Defense.

DSET Dispositivos de Simulação de Engajamento Tático.

EB Exército Brasileiro.

ED Estratégia de Defesa.

EE Estabelecimentos de Ensino.

ELT *Extract-Load-Transform*.

END Estratégia Nacional de Defesa.

EsSA Escola de Sargentos das Armas.

ETL *Extract-Transform-Load*.

F Emp Estrt Forças de Emprego Estratégico.

F Emp Ge Forças de Emprego Geral.

F Esp Emp Estrt Forças Especializadas de Emprego Estratégico.

F Ter Força Terrestre.

FAB Força Aérea Brasileira.

FFAA Forças Armadas.

FLAMES *Flexible Analysis and Mission Effectiveness System.*

FOM *Federation Object Model.*

FORPRON Forças de Prontidão.

FTD *Flight Training Device.*

FTP *File Transfer Protocol.*

GAC Grupo de Artilharia de Campanha.

GAC AP Grupo de Artilharia de Campanha Autopropulsado.

GB *Gigabytes.*

GMF Grupo de Artilharia de Mísseis e Foguetes.

GPU *Graphics Processing Unit.*

GRILL *Gaming Research Integration for Learning Laboratory.*

GUIDEx *Guide for Understanding and Implementing Defense Experimentation.*

HLA *High Level Architecture.*

HTML *HyperText Markup Language.*

HTTP *HyperText Transfer Protocol.*

IA Inteligência Artificial.

IEEE *Institute of Electrical and Electronics Engineers.*

IoT *Internet Of Things.*

IP *Internet Protocol.*

ISR *Intelligence, Surveillance and Reconnaisance.*

JSAF *Joint Semiautomated Forces.*

JSON *JavaScript Object Notation.*

KB *Kilobytes.*

LAN *Local Area Network.*

LC 117/2004 Lei Complementar nº 117, de 2 de setembro de 2004.

LC 136/2010 Lei Complementar nº 136, de 25 de agosto de 2010.

LC 97/1999 Lei Complementar nº 97, de 9 de junho de 1999.

LVC *Live, Virtual and Constructive.*

MB *Megabytes.*

min Minutos.

MIXR *Mixed Reality Simulation Platform.*

ML *Machine Learning.*

Modul Ap Módulos de Apoio.

MPP *Massively Parallel Processing.*

ms Milissegundos.

OEE Objetivos Estratégicos do Exército.

OLAP *Online Analytical Processing.*

OLTP *Online Transaction Processing.*

OM Organizações Militares.

OMT *Object Model Template.*

OND Objetivos Nacionais de Defesa.

OneSAF *One Semiautomated Forces.*

OTAN Organização do Tratado do Atlântico Norte.

PDU *Protocol Data Units.*

Pel Mrt P Pelotão de Morteiro Pesado.

Pel Rec Atq Pelotão de Reconhecimento e Ataque.

PIM Programa de Instrução Militar.

PMT Política Militar Terrestre.

PND Política Nacional de Defesa.

PRE Preocupações Arquiteturais.

RAM *Random Access Memory.*

REOP Reconhecimento, Escolha e Ocupação de Posição.

RF Requisitos Funcionais.

RNF Requisitos Não-Funcionais.

RTI *Run-Time Infraestructure.*

seg Segundos.

SHEFE Simulador de Helicópteros Esquilo/Fennec.

SIMAF Simulador de Apoio de Fogo.

SIPLEx Sistema de Planejamento Estratégico do Exército.

SISOMT Sistema Operacional Militar Terrestre.

SISPREPARO Sistema de Preparo da Força Terrestre.

SISPRON Sistema de Prontidão.

SSD *Solid State Drive.*

SSFTer Sistema de Simulação da Força Terrestre.

SSO *Single Sign-on.*

SVT Simuladores Virtuais Táticos.

SVTat Simulador Virtual Tático.

TB *Terabytes.*

TCP *Transmission Control Protocol.*

TENA *Test and Training Enabling Architecture.*

UDP *User Datagram Protocol.*

USAF *United States Air Force.*

VBS3 *Virtual Batllespace 3.*

WAN *Wide Area Network.*

XML *Extensible Markup Language.*

Capítulo 1

Introdução

O desenvolvimento tecnológico alcança todas as áreas da sociedade. O avanço de tecnologias disruptivas inclui computação avançada, *big data*, *analytics*, Inteligência Artificial (IA), e robótica entre aquelas que são de interesse para a área de defesa [14]. A Organização do Tratado do Atlântico Norte (OTAN) elencou em 2020 a IA como uma das tecnologias emergentes que estão revolucionando as capacidades militares daquela aliança militar com efeitos significativos para o período projetado de 2020 a 2030 [15].

Mais recentemente, o Exército Brasileiro (EB) expediu orientações estratégicas para implantação e uso da IA, com destaque para a integração de sistemas e para as atividades de interesse para o preparo e emprego da Força Terrestre (F Ter) [16]. Apesar da ciência de dados estar presente no setor público e privado, seu uso em aplicações militares de tomada de decisão no nível operacional e estratégico ainda tem sido objeto de poucos estudos acadêmicos [17].

Neste contexto de avanço tecnológico, o EB busca integrar a IA e a Ciência de Dados em suas competências, em especial no preparo da F Ter. As atividades de preparo e desenvolvimento de doutrina militar utilizam sistemas de simulação que produzem uma elevada quantidade de dados, os quais ainda não são processados e analisados em um contexto além do próprio exercício de simulação. Esta situação denota a inexistência de uma infraestrutura de integração que permita a extração de valor completa dos dados gerados, e a inviabilidade da implementação da IA e da Ciência de Dados sem a existência de uma fundação consistente e escalável.

1.1 Justificativa

A Ciência de Dados necessita de uma infraestrutura para existir e não somente dados. Rogati [1] afirma que frequentemente as organizações não estão prontas para a IA por não terem construído a infraestrutura para implementá-la e colher os benefícios disso. Ela

apresenta uma analogia à pirâmide de Maslow [18], na qual as necessidades básicas (engenharia de dados) ficam na base e a utilização dos dados para IA fica no topo. A Figura 1.1, apresenta que as atividades de coleta, movimentação, armazenamento, transformação, categorização, e otimização são fundamentos imperativos para o desenvolvimento da IA e de *analytics*. Reis e Housley [12] destacam que cientistas de dados dedicam de 70% a 80% do tempo às etapas iniciais da pirâmide - coleta, limpeza e processamento de dados. Assim, é essencial construir uma sólida infraestrutura para processar os dados antes de avançar sobre áreas como *analytics* e IA. Essa necessidade também se aplica à simulação de combate, que ainda não possui capacidade de coleta unificada de dados de seus simuladores.

Figura 1.1: Hierarquia de Necessidades de IA (Adaptado de Rogati [1]).

1.2 Objetivos

O presente trabalho tem por objetivo propor uma arquitetura distribuída para integrar dados produzidos por sistemas de simulação no âmbito da F Ter, criando uma infraestrutura que propicia a extração de valor dos mesmos, seja por meio de *analytics* ou IA. Para cumprir este objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Desenvolver uma arquitetura distribuída de integração de dados para simulação de combate;

- Validar a arquitetura por meio da aferição das métricas de seus atributos de qualidade;
- Levantar valores relativos às características de *big data* dos sistemas de simulação: volume, velocidade, variedade, veracidade e valor.

1.3 Organização

Este documento está dividido em capítulos, conforme apresentado a seguir:

- O Capítulo 2 apresenta os fundamentos da simulação militar. A finalidade é conhecer a terminologia inerente ao domínio, compreender os aspectos gerais da simulação distribuída e seus protocolos de integração de simuladores, identificar o contexto normativo do emprego da simulação, e apresentar os principais sistemas em uso no EB;
- O Capítulo 3 ambienta a respeito dos principais conceitos na área da Engenharia de Dados. O objetivo é apreender as principais características do *big data*, conhecer os estágios do ciclo de vida da engenharia de dados e seus aspectos transversais, identificar os atributos de qualidade de software, e apresentar trabalhos relacionados que são referências para o desenho da arquitetura;
- O Capítulo 4 apresenta o desenho, a implementação e a validação do protótipo. Tem por finalidade demonstrar as justificativas para as escolhas arquiteturais, apresentar as tecnologias utilizadas e os detalhes de implementação, e o processo de validação por meio da realização de um experimento para avaliação do protótipo;
- O Capítulo 5 encerra o trabalho, extraindo conclusões obtidas durante o processo de estudo, e inferindo novas direções de estudos futuros.

Capítulo 2

Modelagem e Simulação

Este capítulo tem por objetivo apresentar os conceitos fundamentais em modelagem e simulação, divididos em cinco seções. A Seção 2.1 explica os conceitos fundamentais do domínio. A Seção 2.2 elicita os principais padrões de interoperabilidade de simuladores em uso. A Seção 2.3 mostra a justificativa legal do uso da simulação e apresenta alguns Centros de Instrução (CI) e Centros de Adestramento (CA). A Seção 2.4 demonstra os sistemas de simulação de interesse do trabalho. Por fim, a Seção 2.5 apresenta as conclusões parciais a respeito dos aspectos apresentados.

O conhecimento dos conceitos fundamentais e da situação atual da simulação de combate no âmbito da Força Terrestre (F Ter), fornecem uma contextualização importante para o presente trabalho. É necessário estabelecer um vocabulário comum na área da modelagem e simulação por meio da terminologia e dos conceitos comumente utilizados pela disciplina. Ao mesmo tempo, faz-se necessário identificar o amparo legal e institucional da utilização de simuladores e, consequentemente, fundamentar seu emprego e a forma pela qual a F Ter organiza estes meios. Tais aspectos subsidiam a utilização da tecnologia alinhada a sua finalidade maior, a defesa da Pátria. É importante ressaltar que, para fins do trabalho ora apresentado, os termos simulação de combate e simulação militar são usados indistintamente dentro do contexto da modelagem e simulação, que é o termo internacionalmente utilizado [19].

2.1 Conceitos para Simulação de Combate

O *Department of Defense (DoD)* do Estados Unidos da América definiu em 1998, por meio da publicação *DoD 5000.59-M Modelling and Simulation (M&S) Glossary* [19], os conceitos fundamentais utilizados na área da simulação de combate:

- **Modelo**: representação física, matemática ou lógica de um sistema, entidade, fenômeno ou processo;

- **Simulação**: método de implementação de um modelo ao longo do tempo.

A priori, a simulação é considerada uma ferramenta para planejamento, *design* e avaliação de sistemas dinâmicos, produzindo dados observáveis de suas transformações e mudanças [20]. Entretanto, o *Body of Knowledge for Modeling and Simulation* [21] expande este conceito, considerando a utilização da simulação sob os aspectos da experimentação e da experiência, adicionando o treinamento ao escopo da disciplina:

- **Experimentação**: simulação como sendo a realização de experimentos orientados a objetivos definidos, usando modelos de sistemas dinâmicos;
- **Experiência**: simulação como ferramenta para proporcionar experiência, sob condições controladas, para fins de treinamento.

As chamadas Aplicações da Simulação [13] são uma classificação mais detalhada que a anterior, mas que igualmente apresentam o treinamento como um dos propósitos da área da simulação:

- **Treinamento**: proporcionar oportunidades de aprendizagem e geração de conhecimentos aos instruendos;
- **Análise**: realização de estudos detalhados que apoiam o desenvolvimento de sistemas reais ou imaginados, favorecendo o *design*, teste, avaliação e predição de seu comportamento nos diversos ambientes;
- **Experimentação**: exploração de possibilidades e variação dos resultados, assim como obtenção de *insights* sobre situações de alguma forma incompreensíveis;
- **Engenharia**: realização de testes em diversos *designs* para apoiar o desenvolvimento de sistemas;
- **Aquisição**: especificação, *design*, desenvolvimento e implementação de novos sistemas, incluindo o ciclo de vida dos mesmos, da formulação conceitual ao desfazimento.

O reconhecimento do ganho de experiência como um dos propósitos da simulação torna este último tipo de classificação a mais usada em defesa. Os tipos de simulação são uma taxonomia que deriva da anterior e que consideram a natureza do operador, do sistema e do ambiente, bem como a proficiência a ser treinada, os quais são elencados a seguir:

- **Simulação Virtual**: compreende pessoas reais operando equipamentos ou sistemas simulados em ambiente simulado [22]. É utilizado para melhorar as habilidades motoras para ganho de proficiência no uso de equipamentos [21]. Simuladores de voo ou

simuladores que utilizam cabines, como o Simulador de Helicópteros Esquilo/Fennec (Figura 2.1), são um bom exemplo deste tipo de simulação. Os Simuladores Virtuais Táticos (SVT) são *softwares* de treinamento tático que, ao contrário do mencionado anteriormente, não utilizam periféricos especiais ou que tenham semelhança com o equipamento real. Esses programas funcionam em conjunto com computadores e periféricos comerciais de prateleira, não possuindo *mockups* ou cabines que representem os equipamentos simulados. Entretanto, estes sistemas também estão enquadrados nesta categoria [23], sendo usados para o treinamento dos aspectos cognitivos dos instruendos em seus adestramentos táticos nível pelotão, companhia e batalhão.

Figura 2.1: Simulador SHEFE [2].

- **Simulação Construtiva**: caracterizada por pessoas reais controlando elementos simulados, operando equipamentos ou sistemas simulados em ambiente simulado [22]. O desenvolvimento de habilidades de tomada de decisão e de comunicação são o propósito principal deste tipo de simulação [21]. No caso do EB é empregada como apoio a um Exercício de Posto de Comando [24], onde controladores, auxiliados por operadores, recebem ordens de seus comandos enquadrantes e as lançam no simulador. O sistema processa estas informações e atualiza o estado da simulação. Após isso, as informações de saída são enviadas aos comandos enquadrantes pelos controladores para que sejam emitidas novas ordens. O principal objetivo é proporcionar interações entre os postos de comando em adestramento durante o processo de tomada de decisão e da elaboração dos seus produtos resultantes [25]. Um exemplo deste tipo de simulação é o software Combater, cuja captura de tela pode ser vista na Figura 2.2.

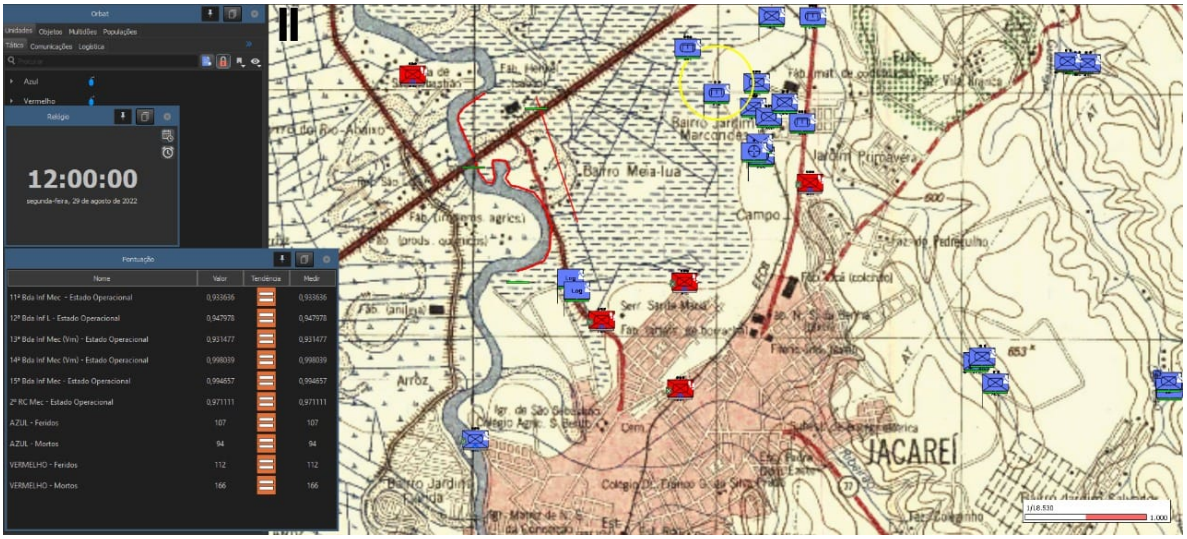

Figura 2.2: Captura de tela do Combater [3].

- **Simulação Viva**: apresenta pessoas reais, operando equipamentos ou sistemas reais no mundo real [22]. O seu propósito é melhorar a capacidade operacional por meio de uma experiência quase real em um ambiente controlado [21]. Os exercícios com apoio de simulação viva podem utilizar sensores, dispositivos apontadores laser e outros instrumentos para acompanhar os instruendos e simular os efeitos dos engajamentos, como ilustrado na Figura 2.3. Isso é realizado por meio do uso dos Dispositivos de Simulação de Engajamento Tático (DSET), que simulam os efeitos dos armamentos com elevada fidelidade permitindo obter dados objetivos dos engajamentos [26]. Um exemplo de soldados utilizando equipamentos de simulação viva pode ser verificado na Figura 2.3.

Figura 2.3: Exemplo de DSET [4].

Além dos tipos de simulação, exitem três atributos fundamentais que ajudam a compreender suas características peculiares. Desta forma, os atributos de simulação de combate, de acordo com Sokolowski e Banks [13], são os seguintes:

- **Fidelidade**: é a qualidade que o modelo ou simulação possui de representar fielmente o sistema real, ou seja, se captura uma quantidade maior ou menor de seus aspectos. Atingir a alta fidelidade não é fácil e, algumas vezes, nem é desejável porque os modelos são construídos para caracterizar somente os aspectos que se quer investigar ou simular;
- **Resolução**: também conhecido como granularidade, é o grau de detalhamento com o qual o mundo real é simulado. Quanto mais detalhes são incluídos na simulação, maior é a resolução;
- **Escala**: também conhecido como nível da simulação, é o tamanho geral do cenário ou evento que ela representa. Quanto maior o cenário, maior a escala. Simulações com elevada escala podem utilizar da agregação, que é a junção de entidades relacionadas em uma única entidade. A maior resolução geralmente leva a menor escala e vice-versa, e isso ocorre porque as simulações são restringidas por limites computacionais.

As Tabelas 2.1 a 2.2 apresentam exemplos que ilustram comparações que ajudam a compreender melhor os atributos da simulação de combate. As mesmas foram adaptadas de Sokolowski e Banks [13].

Tabela 2.1: Comparação entre fidelidade e resolução (Adaptado de Sokolowski e Banks [13]).

	Fidelidade	
Resolução	**Baixa**	**Alta**
Baixa	jogo de xadrez	simulação baseada em agentes
Alta	jogo de simulação de voo para computador pessoal	simulador de voo baseado em plataforma

A simulação militar é utilizada nos diversos escalões de emprego das forças de defesa. Tolk [5] explica que simulações podem ser aplicadas desde o mais alto nível de operações no teatro de guerra, até o mais baixo nível técnico de simulações de propriedades de materiais, como blindagens ou modelos aerodinâmicos e outros equipamentos. Desta forma, o autor declara que os modelos também podem ser classificados de forma hierárquica, compreendendo os chamados níveis de simulação (ver Figura 2.4), elencados a seguir:

Tabela 2.2: Comparação entre fidelidade e escala (Adaptado de Sokolowski e Banks [13]).

	Fidelidade	
Escala	**Baixa**	**Alta**
Alta	jogo de guerra de tabuleiro	jogo de tiro multijogador massivo
Baixa	jogo de tiro para jogador único para celular	jogo de tiro para multijogador

- **Campanha**: no nível estratégico, no qual são empregados modelos de campanhas militares ou de teatro de operações. São utilizados para realizar experimentos na estrutura e no desenho das forças armadas. Geralmente, possuem resolução baixa, o que significa que as muitas entidades são agregadas em um único objeto modelado;
- **Missão**: no nível operacional, no qual missões e batalhas são simuladas para apoiar análises de aplicabilidade doutrinária, planejamento de missões, ou modernização das forças;
- **Engajamento**: no nível tático, quebrando as batalhas e operações militares até o nível de engajamento ou duelo. Visam o melhoramento da tática e sua otimização;
- **Engenharia**: no nível técnico, compreendendo modelos de engenharia. Neste nível, modelos baseados em física e de sistemas reais bem próximos da realidade são utilizados.

Figura 2.4: Níveis de simulação (Adaptado de Tolk [5]).

A presente seção demonstrou os principais conceitos relativos à simulação de combate. É importante ressaltar que o EB utiliza a simulação viva, virtual e construtiva sob o aspecto do treinamento e da experimentação, com fidelidade, resolução e escala variáveis. Na seção seguinte, são apresentados os aspectos da simulação distribuída e seus principais protocolos de integração de simuladores.

2.2 Simulação Distribuída

A Simulação Distribuída é compreendida como uma subárea da área de Sistemas Distribuídos, na qual sistemas computacionais são executados em paralelo por meio de múltiplos computadores autônomos [5]. Sokolowski e Banks [13] declaram que uma simulação distribuída é considerada corretamente executada quando produz os mesmos resultados como se fosse processada sequencialmente em um único processador. Nesta situação, computadores distribuídos se expandem de uma única instalação até uma rede global, frequentemente empregando hardware e software heterogêneo, e tendo uma latência da ordem de centenas de microssegundos até alguns segundos. Os autores também afirmam que o foco da simulação distribuída é a reutilização de modelos por meio da interoperabilidade de componentes de simulação heterogêneos. Além disso, a conexão de componentes geograficamente distribuídos, a interoperabilidade entre sistemas de simulação de fornecedores diferentes, a tolerância a falhas e a proteção de informações sensíveis e propriedade intelectual são benefícios importantes desta tecnologia.

Segundo Tolk [5], os mecanismos de comunicação se referem à abordagem de troca de dados entre dois ou mais simuladores, representando um aspecto importante em simulação distribuída. A transmissão de mensagens apresenta variações em sua forma de entrega, dependendo do número de destinatários. O autor informa que os dados podem ser transmitidos individualmente para simuladores individuais por meio de *unicast*, difundidos por *broadcast* para cada simulação, ou por *multicast* para um subconjunto de simulações. Mecanismos como *publish/subscribe* também podem ser usados para definir subgrupos de destinatários.

A padronização é uma forma de facilitar a interoperabilidade, sendo um objetivo fundamental para todos os *stakeholders*. Desde os primeiros dias da simulação distribuída, os padrões têm tido um papel crucial para a obtenção da interoperabilidade, sendo que os mais utilizados nos dias atuais são [21]: *Test and Training Enabling Architecture* (TENA), *High Level Architecture* (HLA), e *Distributed Interactive Simulation* (DIS); sendo descritos nas próximas seções.

2.2.1 *Test and Training Enabling Architecture* (TENA)

Tolk [5] afirma que o padrão TENA surgiu no final dos anos 1990 com o propósito de prover a arquitetura e a implementação necessárias a interoperabilidade, reuso e composição. A interoperabilidade visa sistemas, instalações, simulações e sistemas C4ISR de forma rápida, e custo-benefício favorável. O autor declara que o reuso busca a ampla reutilização dos meios e vislumbra os desenvolvimentos futuros. A composição compreende a rápida montagem, inicialização, teste e execução de um sistema a partir de um conjunto de elementos reutilizáveis e interoperáveis. Ainda de acordo com o autor, os princípios da arquitetura TENA são os seguintes:

- **Composição limitada**: habilidade de compor o sistema para os propósitos específicos, sejam eles transitórios ou permanentes. As limitações se referem ao uso de meios que incluem proximidade física ou localização, áreas de cobertura, condições de desempenho e compatibilidade dos subsistemas;
- **Caracterização dinâmica de tempo de execução**: capacidade de resposta a muitas composições permitidas e sua rápida reconfiguração. Isto é realizado pelo estabelecimento de métodos para auto descrição de representações de dados anteriormente ou de forma concorrente com a transferência dos dados, ou negociando problemas de representação antes da operação do sistema;
- **Subscrição em serviços**: abordagem baseada em objetos para acesso de dados, os quais combinam produtores e consumidores de informação;
- **Acesso controlado à informação**: por meio de níveis de acesso que limitam o ingresso no sistema para um subgrupo de usuários, favorecendo quanto às implicações de desempenho e custos. Os usuários podem solicitar alocação de meios especiais quando necessário;
- **Qualidade de serviço negociada**: os protocolos dependem do princípio de separação entre controle da informação e dados.

O TENA *middleware* é um mecanismo de comunicação comum a todas as aplicações, sendo uma solução única e universal de troca de dados. Ele combina alguns paradigmas de comunicação que incluem: memória distribuída compartilhada, *publish-subscribe* anônimo, evocação remota de métodos e suporte nativo para *streams* de dados. Os protocolos de comunicação de rede utilizados podem ser tanto o UDP quanto o TCP. O TENA é gerenciado por autoridades governamentais americanas e não há padrões abertos publicados [5]. Embora o TENA seja amplamente usado em contextos internacionais, sua aplicação no EB é desconhecida, entretanto o padrão serve como referência de interoperabilidade entre simuladores.

2.2.2 *High Level Architecture* (HLA)

Tolk *et al.* [21] explicam que o padrão HLA teve início em 1995 por meio de um processo governamental de padronização. O DoD adotou a arquitetura em 1996 e, em 2000, foi aprovada a primeira versão das normas IEEE 1516, relativas ao padrão, sendo as mesmas atualizadas em 2010. Ainda segundo os autores, a arquitetura HLA é definida por três componentes:

- ***Object Model Template*** **(OMT)**: definição e especificação de um modelo comum;
- **Especificação de Interface**: serviços que descrevem o ambiente de execução;
- **Regras do HLA**: regras de conformidade com a arquitetura.

O HLA foi desenvolvido para mitigar a proliferação de soluções de integração de simuladores em sua época. Desta forma, permite a sua aplicação em um grande rol de sistemas de simulação para apoiar o treinamento, ensaio de missão, análise, teste e avaliação. Para entender o HLA, alguns conceitos importantes são necessários, ainda segundo os autores:

- **Federação**: é uma coleção de federados (simuladores e outros sistemas) que são integrados usando os protocolos da arquitetura;
- ***Run-Time Infraestructure*** **(RTI)**: serviços comuns que permitem a comunicação eficiente entre os federados, por meio da separação entre as funcionalides dos simuladores da infraestrutura necessária para comunicação. É a implementação dos serviços definidos na especificação de interface;
- ***Federation Object Model*** **(FOM)**: é a instância do OMT provendo a especificação do modelo e o estabelecimento do contrato entre os federados. Eles indicam as informações que são fornecidas a federação e aquelas que são aceitas da federação.

A arquitetura HLA, segundo Tolk [5], utiliza o paradigma *publish/subscribe*. Esta abordagem resulta em máximo desempenho na rede, onde sistemas de simulação podem filtrar em muitos níveis diferentes os dados que querem receber. O autor afirma que a arquitetura apresenta serviços de gestão de tempo e de ordem de eventos. No primeiro caso, as mensagens são entregues seguindo a ordem temporal por meio do *timestamp*. No segundo caso, as mensagens são entregues conforme a ordem recebida.

O padrão é amplamente utilizado para conectar simulações usando o RTI como *middleware*, segundo Sokolowski e Banks [13]. Entretanto, os autores levantam algumas questões observadas quando ao seu uso. Em primeiro lugar, não é possível assumir que há interoperabilidade entre RTI de diferentes fornecedores. Isso ocorre porque não existem padrões para a implementação do RTI, nem para a comunicação entre eles. Em segundo

lugar, o sistema não escala bem quando há muitas simulações conectadas no mesmo RTI. Em terceiro lugar, as especificações de API estão unidas a linguagens de programação. Por último, os autores afirmam que os *firewalls* geralmente bloqueiam a comunicação do RTI quando usadas na *Internet* ou em *Wide Area Network* (WAN).

2.2.3 *Distributed Interactive Simulation* (DIS)

O padrão DIS tem por finalidade conectar diferentes tipos de simuladores em múltiplos locais, sendo regulado pela norma IEEE 1278 publicada pela primeira vez em 1993, segundo Tolk [5]. A característica principal deste padrão são os *Protocol Data Units* (PDU), que são mensagens padronizadas transmitidas entre os sistemas de simulação e que comunicam o estado de entidades e eventos. A implementação de sua interface permite tanto a aquisição de produtos comercialmente disponíveis quanto o desenvolvimento de interfaces customizáveis, com destaque para a iniciativa *open source* chamada Open-DIS [27]. O autor explica que toda a comunicação sobre o estado das entidades e suas interações ocorre por meio do PDU sendo que a entrega das mensagens é confiável, com menos de 2% dos datagramas perdidos. Apresenta latência alta nas WANs, sendo recomendado seu uso em redes locais ou *Local Area Network* (LAN), e os PDUs são distribuídos na rede por meio de *broadcast* das mensagens. Desta forma, ainda segundo o autor, os recursos computacionais e de rede são consumidos com dados que não são relevantes para cada simulador.

A norma IEEE 1278 define o formato e a semântica das mensagens fornecendo informações sobre o estado das entidades simuladas, os tipos de interações entre elas no exercício, o gerenciamento e controle do exercício, os estados simulados do ambiente, a agregação de entidades e a transferência de posse das entidades [28]. É importante ressaltar que a implementação *Open-DIS* é gratuita e de código aberto nas linguagens Java, C++, Python, Javascript, Objective–C e C# [27]. No caso da implementação em C#, o *Gaming Research Integration for Learning Laboratory* (GRILL), da Força Aérea Americana, desenvolveu um *plugin* para os motores de jogos digitais *Unity* [29] e *Unreal* [30] que realiza a implementação do protocolo usando o *Open-DIS* como base, sendo os mesmos disponibilizados publicamente no *GitHub* [31].

2.3 Emprego da Simulação

Os padrões de interoperabilidade entre simuladores abrem um leque de possibilidades de seu emprego. A simulação de combate, principal vetor de preparação de qualquer força armada, deve alinhar-se aos pressupostos legais que orientam o preparo, justificando a utilização desta tecnologia. Neste contexto, é importante conhecer a priorização elencada

pelo EB para o emprego dos meios de simulação, assim como identificar a distribuição geográfica das instalações possuidoras de simuladores, facilitando a visualização inicial dos desafios para integração de dados dos mesmos. As seções seguintes apresentam os documentos estratégicos norteadores da atividade, o processo de certificação de tropas, aspectos da experimentação doutrinária e um resumo da situação atual no âmbito da F Ter.

2.3.1 O Preparo da F Ter

A Constituição da República Federativa do Brasil de 1988 (CRFB) [32] é o principal documento norteador que define a missão das Forças Armadas (FFAA), especificando como sua principal incumbência a defesa da Pátria. O cumprimento dessa missão constitucional tem sua diretriz de mais alto nível apresentada na Lei Complementar nº 97, de 9 de junho de 1999 (LC 97/1999) [33], alterada posteriormente pela Lei Complementar nº 117, de 2 de setembro de 2004 (LC 117/2004) [34], nelas as forças singulares recebem a responsabilidade pelo preparo de seus órgãos operativos e de apoio. Esta atividade é realizada por meio de ações entre as quais se encontram a instrução, o adestramento, o desenvolvimento da doutrina e pesquisa, e atividades de inteligência.

A LC 97/1999 também foi alterada pela Lei Complementar nº 136, de 25 de agosto de 2010 (LC 136/2010) [35] que instituiu a Política Nacional de Defesa (PND) e a Estratégia Nacional de Defesa (END), a primeira estabelece os Objetivos Nacionais de Defesa (OND), e a segunda orienta as medidas a serem implementadas para atingir estes objetivos. As ações de preparo estão ligadas, notadamente, ao alinhamento estratégico caracterizado pelo OND-II: Assegurar a Capacidade de Defesa para o Cumprimento das Missões Constitucionais da Forças Armadas, em sua Estratégia de Defesa (ED)-6: Capacitação e Dotação dos Recursos Humanos, materializada pela Ação Estratégica de Defesa (AED)-29: Manter os Efetivos Adequadamente Preparados. O estabelecimento de objetivos estratégicos institucionais sintetiza a intenção de manter tropas militares permanentemente prontas, proporcionando as capacidades necessárias para a defesa do território.

No âmbito do EB, a instituição executa as diretivas superiores utilizando-se de um arcabouço de documentos de planejamento de alto nível, denominado Sistema de Planejamento Estratégico do Exército (SIPLEx), que se encontra, atualmente, abrangendo o horizonte temporal entre 2024 a 2027. Um destes documentos é a Política Militar Terrestre (PMT) [36] que estabelece os Objetivos Estratégicos do Exército (OEE) dos quais se destaca o OEE 4: Aperfeiçoar o Sistema Operacional Militar Terrestre (SISOMT), sistema este responsável pela preparação e geração de forças para emprego operacional, cujo órgão de direção central é o Comando de Operações Terrestres (COTER). Um dos integrantes do SISOMT é o Sistema de Preparo da Força Terrestre (SISPREPARO) que

tem por objetivo principal a formação da reserva mobilizável e o adestramento da F Ter, dispondo de meios de simulação para a qualificação do pessoal integrante das Organizações Militares (OM). Outro documento integrante do SIPLEx é a Concepção Estratégica do Exército (CEE) [37] que estabelece as prioridades do preparo da F Ter como sendo as Forças de Emprego Estratégico (F Emp Estrt), as Forças Especializadas de Emprego Estratégico (F Esp Emp Estrt) e os Módulos de Apoio (Modul Ap). Adicionalmente, atribui ao COTER, por meio do Sistema de Prontidão (SISPRON) apoiado pelo Sistema de Simulação da Força Terrestre (SSFTer) a geração das capacidades operacionais necessárias ao preparo da F Ter por meio da realização de ciclos completos de preparação das tropas priorizadas.

O COTER, no alcance de suas atribuições, batizou as forças componentes do SISPRON de Forças de Prontidão (FORPRON) [38] e, por meio do Programa de Instrução Militar (PIM), particularmente em sua edição 2024 [39], detalhou a composição das forças elencadas na CEE. O exame da legislação de mais alto nível que regula as atividades de preparo da F Ter demonstra o alinhamento estratégico e a sua importância no contexto da Defesa Nacional. É possível verificar que o EB estabeleceu uma prioridade de treinamento e que o SSFTer apoia esta preparação, também priorizando seus meios para este esforço. Em seguida, é apresentada uma visão geral do Ciclo de Prontidão e de sua principal atividade de preparo, a certificação.

2.3.2 A Certificação das FORPRON

A Diretriz Organizadora do Sistema de Prontidão Operacional da Força Terrestre (F Ter), publicada em 2019 [38], estabelece a constituição, o adestramento e a sustentação das FORPRON. Quanto ao adestramento, ela define um ciclo de preparo específico denominado prontidão operacional, que é o estado final desejado no qual a tropa selecionada, adestrada e certificada permanece em condições de, ao ser acionada, deslocar-se a uma área designada dentro de um prazo limitado para cumprir suas missões designadas. O PIM 2024 [39], atualizando algumas das diretivas de preparo das FORPRON, determina que cada Brigada da F Emp Estrt e da F Emp Ge possua organização fixa, sendo composta, pelo Comando e Estado-Maior da Brigada, 1 (uma) unidade nível Batalhão completa com 3 a 4 (três a quatro) subunidades de manobra e 1 (uma) subunidade de comando e apoio, 1 (um) Esquadrão de Cavalaria a 1 (um) pelotão, 1 (um) Grupo de Artilharia de Campanha a 1 (uma) Bateria de Obuses, 1 (uma) Bateria de Artilharia Antiaérea a 1 (uma) Seção de Artilharia Antiaérea, 1 (uma) Companhia de Comunicações, e 1 (um) Destacamento Logístico. Desta forma, estabelecendo o universo de instruendos a ser preparado em cada Brigada.

O Ciclo de Prontidão abrange três fases distintas: preparação, certificação e prontidão. A primeira, com duração de 3 (três) meses, tem por objetivo capacitar o pessoal nas instruções individuais básicas e de qualificação, assim como realizar exercícios de nível Pelotão e Subunidade [39]. A segunda fase, a de certificação, tem a duração de 3 a 4 (três a quatro) semanas para avaliar a preparação da tropa por meio dos sistemas de simulação e com o apoio dos Centros de Adestramento (CA). Nela são realizados exercícios com apoio de simulação construtiva, virtual e viva, além dos exercícios nos dois sistemas de simulação do tipo Simulador de Apoio de Fogo (SIMAF). Nestas ocasiões, os diversos escalões realizam planejamentos e execuções de ações militares enquadradas em situações táticas e, ao final, o Comando Militar de Área (C Mil A) informa ao COTER os efetivos certificados e aptos para ingressarem na próxima fase [39]. A última fase compreende a prontidão propriamente dita, é dividida em dois níveis, o primeiro com 12 (doze) meses de duração e o segundo com 8 (oito) meses de duração, no qual as tropas certificadas poderão ser acionadas para emprego imediato. Nesta fase também há a previsão da participação em exercícios de adestramento para manutenção de padrões, ou integrando contingentes em exercícios conjuntos ou combinados, assim como adestramentos avançados [39].

Os Módulos Estratégicos Especializados, que é a junção das Forças Especializadas de Emprego Estratégico (F Esp Emp Estrt) e dos Módulos de Apoio (Modul Ap), realizam seu Ciclo de Prontidão de forma semelhante às Brigadas das Forças de Emprego Estratégico (F Emp Estrt) e das Forças de Emprego Geral (F Emp Ge) prioritárias. Neste caso, o Ciclo de Prontidão compreende três fases: certificação técnica, prontidão operacional e certificação tática, com a realização de um único exercício por ano, sem apoio de simulação, e tendo o próprio comando enquadrante como autoridade certificadora [39].

O COTER realiza o planejamento anual do preparo por meio do PIM, definindo direção geral da instrução e do adestramento de toda a F Ter, em particular das tropas prioritárias já citadas. Neste contexto, a Certificação das FORPRON é a atividade com grande emprego de sistemas de simulação, sendo importante destacar o apoio de simulação prestado pelos Centros de Adestramento (CA) e Centros de Instrução (CI), assim como Estabelecimentos de Ensino (EE). No âmbito deste trabalho merecem destaque os seguintes:

- **Centro de Adestramento - Sul (CA-SUL)**: tem por missão contribuir para o adestramento de tropas de qualquer natureza, preferencialmente, blindadas e mecanizadas, bem como certificar estas tropas utilizando meios de simulação de combate [40]. Localiza-se em Santa Maria, RS;
- **Centro de Adestramento - Leste (CA-LESTE)**: tem por missão conduzir o adestramento de tropas por meio do emprego da simulação de combate [41]. Fica situado no Rio de Janeiro, RJ;

- **Centro de Instrução de Blindados (CI Bld)**: tem por missão capacitar, especializar ou estender o conhecimento de oficiais e sargentos das FFAA em relação aos blindados nas vertentes técnica, tática e logística até o escalão subunidade [42]. É localizado em Santa Maria, RS;
- **Centro de Instrução de Aviação do Exército (CIAvEx)**: tem como atribuições formar e especializar recursos humanos e contribuir para a evolução da doutrina da Aviação do Exército [43]. É localizado em Taubaté, SP;
- **Centro de Instrução de Artilharia de Mísseis e Foguetes (CI Art Msl F)**: tem por missão especializar os recursos humanos no emprego e na logística do sistema de mísseis e foguetes, assim como contribuir para a formulação da sua doutrina de emprego [44]. Localiza-se em Formosa, GO;
- **Academia Militar das Agulhas Negras (AMAN)**: tem por missão formar os oficiais combatentes de carreira [45]. É situada em Resende, RJ.

2.3.3 Experimentação Doutrinária

O EB, em legislação específica [46], define Experimentação Doutrinária como o conjunto de atividades com o objetivo de validar a exequibilidade e a eficácia de conceitos, técnicas, procedimentos e estruturas do emprego da F Ter. Ela é caracterizada por um experimento de campo realizado sob condições que se aproximem ao máximo da realidade, utilizando militares e equipamentos que geram documentos doutrinários que servem de base teórica futura. A legislação também apresenta a simulação de combate como ferramenta a ser utilizada para extrair conclusões que permitem o melhor entendimento do comportamento do sistema a ser experimentado, sendo possível a aplicação dos conceitos adquiridos em seus correspondentes no mundo real.

O DoD apresenta, em normatização própria [47], a experimentação como a aplicação do método científico para determinar relações de causa e efeito. Em um ambiente controlado, é realizada a manipulação de uma ou mais entradas e o registro de seus efeitos na saída, sendo que no final é realizada a análise dos dados para validar os relacionamentos encontrados. A norma também afirma que a experimentação de defesa é uma extensão do pensamento experimental para o domínio militar, sendo definida pelo teste de hipóteses, sob condições controladas, para explorar efeitos desconhecidos da manipulação de conceitos, tecnologias ou condições de guerra.

O *Guide for Understanding and Implementing Defense Experimentation* (GUIDEx) [48], foi elaborado por Austrália, Canadá, Reino Unido e Estados Unidos, para consolidar os princípios e diretrizes para a melhoria do impacto da experimentação no desenvolvimento da capacidade militar. O GUIDEx apresenta 14 princípios para orientar os países

aliados no emprego da experimentação em defesa, e orienta o emprego da simulação em dois deles. No primeiro, recomenda o uso de múltiplos métodos para conduzir experimentos incluindo jogos de guerra, simulações construtivas, simulações *human-in-the-loop* e simulação viva (ou experimentos de campo). No segundo, enfatiza que a exploração da simulação de combate é crítica para o sucesso da experimentação e estima que mais de oitenta porcento dos experimentos em defesa utilizam simulação de alguma forma.

O Exército Americano reconhece a importância da experimentação baseada em simulação. O novo conceito de Operações Multidomínio [49] apresenta um ambiente operacional que é compreendido por porções dos domínios terrestre, marítimo, aéreo, espacial e cibernético. Estas partes são compreendidas por meio de suas dimensões humana, física e informacional. Assim, este ambiente considera a totalidade de fatores, circunstâncias específicas e condições que impactam as operações. O relatório elaborado pelo *Army Science Board* [50] afirma que o Exército Americano precisa ser capaz de credibilizar a efetividade em combate de suas forças em tão complexo ambiente operacional. Para isso, deve realizar experimentações com seus sistemas em um construto conjunto para avançar a doutrina, e desenvolver uma arquitetura de sistemas multidomínio.

2.3.4 Situação Atual

A simulação de combate apresenta duas vertentes importantes de utilização. A primeira é o preparo da F Ter, a qual é empregada diretamente no treinamento e preparação das tropas, com ênfase para as FORPRON, para seu emprego operacional. Neste contexto, o apoio dos CA e CI é caracterizado pela manipulação dos meios de simulação e pela dedicação de seu pessoal especializado.

A segunda vertente é a experimentação de defesa, onde a simulação de combate apoia o teste de hipóteses doutrinárias, oferecendo o lastro teórico e prático necessário para o desenvolvimento de capacidades militares. É importante lembrar que os exercícios militares, seja no âmbito do preparo ou da experimentação, são atividades estanques no tempo e espaço quando observados no conjunto dos demais exercícios, não necessitando de acompanhamento em tempo real.

2.4 Sistemas em Uso no EB

O EB emprega a simulação virtual, viva e construtiva, por meio dos seus CI e CA em todas as fases do Ciclo de Prontidão das FORPRON, e nas experimentações doutrinárias. É importante realizar a correspondência entre o simulador utilizado, sua localização geográfica e suas possibilidades de interoperabilidade ou integração.

2.4.1 Simulação Virtual

A simulação virtual representa a maioria dos sistemas em uso pelo EB nos dias atuais. As cabines de simulação e os SVT são empregados na qualificação de pessoal e nos treinamentos coletivos. Os principais sistemas em utilização são os seguintes:

- **Software de Simulação *Virtual Batllespace 3* (VBS3)**: é produzido pela empresa *Bohemia Interactive* que reproduz cenários, armamentos, equipamentos e viaturas em um ambiente virtual replicado por meio do computador, sendo empregado no treinamento desde o nível individual até o nível coletivo [51] (ver Figura 2.5). O sistema é utilizado no CA-SUL, CA-LESTE, CI Bld, assim como na AMAN e Escola de Sargentos das Armas (EsSA) [52]. O VBS3 possui capacidade de conexão DIS ou HLA com outros simuladores por meio do VBS *Gateway*, que é fornecido por padrão juntamente com a licença do sistema [53].

Figura 2.5: Militar realizando treinamento no VBS3 [6].

- **Software de Simulação Virtual *Steel Beasts***: é um programa de computador desenvolvido pela empresa *eSim Games*, o qual representa as interações entre as tropas blindadas e mecanizadas em ambiente virtual (ver Figura 2.6). Diferentemente do VBS3, ele é concebido para tropas embarcadas, apresentando maior realismo nos engajamentos entre os blindados. É empregado pelo CI Bld [54], e utiliza o protocolo de integração DIS [55].
- **Simulador Virtual Tático (SVTat) REOP**: é um sistema que tem por objetivo adestrar os militares especializados em artilharia de mísseis e foguetes no treinamento do Reconhecimento, Escolha e Ocupação de Posição (REOP) do Grupo de Artilharia de Mísseis e Foguetes (GMF). Apresenta ambiente 3D mostrado em tela e em uma mesa tática visando a melhor visualização das decisões tomadas (ver Figura

Figura 2.6: Captura de tela do *Steel Beasts [7].*

2.7). Está instalado no CI Art Msl F [56]. Ele é desenvolvido por meio da utilização do motor de jogos digitais *Unity* [57].

Figura 2.7: Demonstração do SVTat REOP [8].

- **Simulador de Apoio de Fogo (SIMAF)**: é um sistema complexo de simulação para Grupo de Artilharia de Campanha (GAC) e Pelotão de Morteiro Pesado (Pel Mrt P) fabricado pela empresa espanhola *Tecnobit*. Este simulador tem por finalidade apoiar o treinamento das capacidades operativas das frações de apoio de fogo, misturando elementos de linha de fogo, central de tiro e observação para realizar o adestramento tático (ver Figura 2.8). O EB possui dois simuladores instalados, sendo um em Resende - RJ, na AMAN, e outro em Santa Maria - RS, no CA-SUL [58]. A versão 2.0, está em desenvolvimento pelo Centro de Desenvolvimento de Sistemas (CDS), utiliza o motor de jogos digitais *Unity*.

Figura 2.8: Posto de observação do SIMAF [9].

- **Software de Simulação Virtual Bombarda**: é um sistema desenvolvido com o objetivo de auxiliar na formação e no adestramento de Observadores Avançados, além do adestramento integrado de subsistemas de artilharia (ver Figura 2.9). É um software utilizado por OM fora do contexto dos CI e CA, sendo que as OM que atualmente o utilizam são: o 14º Grupo de Artilharia de Campanha (GAC) (Pouso Alegre, MG) [59], o 32º Grupo de Artilharia de Campanha (GAC) (Brasília, DF) [10], o 5º Grupo de Artilharia de Campanha (GAC) (Curitiba, PR) [60], o 7º Grupo de Artilharia de Campanha (GAC) (Olinda, PE) [61], o 28º Grupo de Artilharia de Campanha (GAC) (Criciúma, SC) [62], o 19º Grupo de Artilharia de Campanha (GAC) (Santiago, RS) [63], e o 15º Grupo de Artilharia de Campanha Autopropulsado (GAC AP) (Lapa, PR) [64]. Ele foi desenvolvido usando o motor de jogos digitais *Unity* [65].

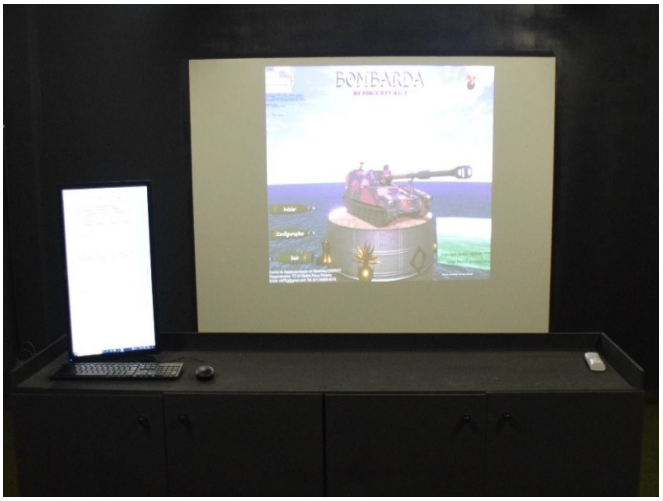

Figura 2.9: Exemplo de uma sala Bombarda [10].

- ***Flight Training Device* (FTD)**: segundo a Agência Nacional de Aviação Civil (ANAC), consiste em uma réplica dos instrumentos, equipamentos, painéis e controles de uma aeronave confinada ou não na cabine de pilotagem, seus equipamentos e programas necessários, e sem sistema da movimento [66]. O CIAvEx possui cabines de FTD e estações de controle que permitem voo visual, voo por instrumentos, voo com óculos de visão noturna, voo em formação tática e operações aeromóveis, favorecendo o adestramento de Pelotão de Reconhecimento e Ataque (Pel Rec Atq) orgânico da AvEx (ver Figura 2.10). O software utilizado para a simulação é o *X-Plane* [67], e a implementação de uma interface de integração DIS é facilitada [68].

Figura 2.10: Exemplo de uma cabine FTD [11].

2.4.2 Simulação Construtiva

O software de simulação construtiva Combater é desenvolvido pela empresa francesa MASA, sendo baseado no software *Sword* e customizado às peculiaridades da Doutrina Militar Terrestre brasileira. Este software compreende um sistema de simulação que emprega um cenário digital dedicado a adestrar os postos de comando de batalhões, brigadas e divisões, permitindo o exercício das ações de planejamento, coordenação e tomada de decisão. O software tem a capacidade de simular cenários de guerra simétrica ou assimétrica, além de segurança pública e cooperação e coordenação interagências [69]. Tanto o CA-LESTE, quanto o CA-SUL utilizam o Combater na simulação construtiva [70]. Possui interface de integração HLA que permite conexão a federações que utilizem esta arquitetura [71], além de ser possível desenvolver um cliente remoto usando o *Google Protocol Buffers* por meio da API do sistema [72].

2.4.3 Simulação Viva

Os Dispositivos de Simulação de Engajamento Tático (DSET) são equipamentos de simulação viva que utilizam a tecnologia do laser e do rádio para simular, com fidelidade, os efeitos dos diversos tipos de armamento e outros dispositivos, modelando um cenário de combate ao mesmo tempo que não incorre em danos reais aos envolvidos [26]. Atualmente, o EB utiliza os equipamentos produzidos pela empresa sueca *SAAB Training and Simulation*, e tanto o CA-LESTE [73] quanto o CA-SUL [74] empregam esta tecnologia em seus exercícios de adestramento. O sistema em uso permite integração DIS/HLA [75].

2.4.4 Sistemas em Uso no EB

Os sistemas em uso pelo EB se caracterizam pelas muitas aplicações e meios para apoiar a F Ter. Eles possuem compatibilidade com os principais protocolos de integração de simuladores ou protocolos de comunicação abertos, como o *Google Protocol Buffers*, o que permite o desenvolvimento de interfaces *in-house*. Os muitos meios existentes estão distribuídos em todo o território nacional, devendo os CA e CI estarem integrados dentro de uma arquitetura que atenda a esta particularidade.

A Tabela 2.3 apresenta um resumo da situação atual dos simuladores elencando os sistemas existentes, o protocolo de comunicação disponível, se há implementação nativa do lado da aplicação, e as unidades militares onde estão instalados. Aqueles sistemas que não possuem implementação de comunicação nativa do lado da aplicação podem ser alvo de desenvolvimento posterior, como já particularizado nas seções anteriores.

2.5 Conclusões Parciais

Este capítulo apresentou as principais características da simulação de combate utilizados em proveito da F Ter. O EB possui sistemas de simulação de todos os tipos (virtual, construtiva e viva), os quais utilizam protocolos de integração padronizados, sendo possível desenvolver interfaces customizadas para atender a comunicação com estes sistemas, possibilitando a coleta dos dados dos mesmos. O preparo e a experimentação doutrinária são fortemente apoiados pelos meios de simulação existentes, o que potencializa o treinamento e respaldo aos experimentos doutrinários.

Os principais vetores da simulação, os CA e CI, estão localizados nos diversos rincões do país e a coleta de dados dos exercícios pode ser feita periodicamente, de forma agendada, pois a análise dos resultados dos exercícios, em um contexto mais amplo, é realizada *a posteriori*. A situação atual apresentada coloca o CA-SUL e o CA-LESTE como os candidatos adequados para a implementação inicial por possuírem sistemas de

Tabela 2.3: Simuladores em uso pelo EB.

Simulador	**Protocolo**	**Nativo**	**OM**	**Localização**
VBS3	DIS	✔	**CI Bld**	Santa Maria, RS
			CA-SUL	Santa Maria, RS
			CA-LESTE	Rio de Janeiro, RS
Steel Beasts	DIS	✔	**CI Bld**	Santa Maria, RS
SVTat REOP	DIS	✖	**CI Art Msl F**	Formosa, GO
SIMAF	DIS	✖	**CA-SUL**	Santa Maria, RS
			AMAN	Resende, RJ
Bombarda	DIS	✖	várias	várias
FTD	DIS	✖	**CIAvEx**	Taubaté, SP
Combater	*Protobuf*	✔	**CA-SUL**	Santa Maria, RS
			CA-LESTE	Rio de Janeiro, RS
DSET	DIS	✔	**CA-SUL**	Santa Maria, RS
			CA-LESTE	Rio de Janeiro, RS

Legenda: ✔= Possui, ✖= Não Possui

simulação com capacidade de comunicação nativa, nos três tipos de simulação (virtual, construtiva e viva). As características do emprego da simulação apresentadas devem ser consideradas na elaboração de uma arquitetura que integre todos estes sistemas.

Capítulo 3

Engenharia de Dados

Este capítulo tem por finalidade exibir conceitos relativos à engenharia de dados, sendo estruturado em cinco seções. A Seção 3.1 ambienta sobre os principais conceitos em *Big Data* (BD). A Seção 3.2 demonstra os aspectos do ciclo de vida da engenharia de dados. A Seção 3.3 conceitua os atributos de qualidade de software. A Seção 3.4 apresenta os estudos que embasaram a formulção da solução. Finalmente, a Seção 3.5 conclui com as anotações finais sobre o capítulo apresentado.

O capítulo anterior demonstrou que o Exército Brasileiro (EB) emprega um rol de sistemas de simulação que estão distribuídos geograficamente pelo território nacional. Estes sistemas produzem dados que podem ser aproveitados tanto no contexto isolado de cada exercício, quanto de forma mais ampla para cooperar com a geração de capacidades militares. Desta forma, a aplicação de conceitos e soluções em BD e engenharia de dados tem o potencial de subsidiar uma arquitetura de integração capaz de extrair valor dos dados produzidos pela simulação de combate.

3.1 Conceitos em Big Data

O objetivo principal da integração de dados é a extração de valor. Os dados são a informação bruta que é manipulada e processada para produzir inteligência de negócio. Assim, os Tipos de Dados são os seguintes [76]:

- **Estruturados**: dados que estão de acordo com regras claramente definidas sobre sua forma e conteúdo. Bancos de dados são formados por dados estruturados e forçam a aplicação destas regras, evitando a entrada de dados inesperados;
- **Semiestruturados**: dados que são formatados de acordo com certas regras aceitas mas que podem variar em sua estrutura. São exemplos o *JavaScript Object Notation*

(JSON), o *Extensible Markup Language* (XML) e as páginas *HyperText Markup Language* (HTML);

- **Desestruturados**: dados que não seguem nenhum formato específico. São exemplos: postagens em *blogs*, mensagens em *emails*, arquivos de vídeo e áudio, e comentários na rede social.

Elmasri e Navathe [77] apresentam o termo *Big Data* (BD) como conjuntos de dados cujo tamanho está além das capacidades de captura, armazenamento, gerenciamento e análise das ferramentas típicas de sistemas de bancos de dados. Adicionalmente, afirmam que a noção de BD tem dependência direta com o tipo de indústria, a forma como os dados são utilizados e seu histórico, além de outras características. Jukic *et al.* [78] também referem BD à conjuntos de dados diversos de volume massivo e de crescimento rápido, que não são formalmente modelados para consultas e nem possuem metadados vinculados. Eles consideram que os conjuntos de dados exibem grande volume, velocidade e variedade, alta probabilidade de problemas de qualidade (veracidade), elevada possibilidade de interpretações diferentes (variabilidade), necessitam abordagem mais exploratória e experimental para extrair valor e, provavelmente, necessitam beneficiar-se de visualizações ricas e inovadoras. Os autores afirmam, ainda, que a literatura apresenta o BD em termos de suas características, também conhecidas como um número variado de 'Vs', que no caso dos autores mencionados, são os seguintes:

- **Volume**: o grande volume ocupado pelos conjuntos de dados;
- **Variedade**: a abundância das fontes de diferentes tipos de dados;
- **Velocidade**: a grande velocidade dos dados de entrada em seu repositório;
- **Veracidade**: os problemas quanto à qualidade dos dados;
- **Variabilidade**: as possibilidades de diferentes interpretações relativas aos dados;
- **Valor**: a utilidade e praticidade das informações extraídas;
- **Visualização**: a necessidade de visualizar os grandes conjuntos de dados.

Song *et al.* [79] apresentam um estudo sobre o BD em simulação militar que revelam alguns dados iniciais na correlação entre as duas áreas do conhecimento. Entre os casos de uso mapeados, encontram-se exercícios conjuntos de experimentação, e emprego da simulação em análise de linhas de ação. No primeiro caso, o DoD realizou uma série de exercícios de larga escala, denominados *Urban Resolve*, com o intuito de desenvolver táticas e avaliar novos sistemas de armas em ambiente urbano, tudo por meio do sistema *Joint Semiautomated Forces* (JSAF). Em uma primeira fase foram simuladas mais de

100.000 entidades, em sua maioria de civis, em centenas de nós executando o JSAF conectados em vários locais distantes geograficamente, sendo coletado 3.7 TB de dados dos modelos por exercício. Em exercícios posteriores foram utilizadas dezenas de milhões de entidades e modelos de resolução mais alta.

No segundo caso, os autores informam que o Exército Americano utilizou o sistema *One Semiautomated Forces* (OneSAF) na avaliação de planos possíveis e múltiplos pontos de decisão, o que é conhecido como análise de linhas de ação baseado em simulação. Este tipo de simulação é executada em tempo mais-que-real, na qual a simulação busca prever o comportamento de sistemas ultra complexos por meio de medições e atributos, e utiliza menos tempo do que o passo de tempo de uma simulação em tempo real [80]. Song *et al.* [79] acrescentam que a simulação em tempo mais-que-real, emprega baixa resolução com grande quantidade de entidades, ou resolução mais alta para uma análise mais profunda e detalhada. O OneSAF possui a seguintes capacidades: modelagem multirresolução, modelagem baseada em agentes, simulação de 33.000 entidades no nível brigada, escalabilidade horizontal, e integração com sistemas reais, podendo gerar dados massivos com a finalidade de analisar e comparar as linhas de ação.

As informações colhidas pelos autores do estudo e que são relevantes para a presente proposta encontram-se sumarizadas nas Tabelas 3.1 a 3.2. Com relação ao volume e velocidade, observaram que o tamanho dos dados chega ao nível de GB a TB por exercício, sendo considerado menor se comparado com o de atividades comerciais, ou o encontrado em redes sociais. Reconhecem que o volume de dados continua a aumentar porque a simulação militar está crescendo, com exercícios de maior escala e resolução, sendo impulsionada pela evolução dos computadores de alto desempenho e pela necessidade de geração e análise de dados em tempo-mais-que-real. Com relação à variedade, verificaram os vários tipos de dados usados em simulação militar que incluem as saídas geradas pelos modelos computacionais, assim como dados desestruturados (como *logs* de simuladores), dados semiestruturados (como dados de entrada e configuração de cenários de simulação) e dados estruturados (como tabelas de bancos de dados).

Com relação à veracidade, afirmam que os dados gerados podem estar incorretos somente se o modelo estiver incorreto, o que requer atividades de verificação e validação dos modelos empregados. Outro aspecto citado, foi a constatação de que a fidelidade dos modelos de simulação é considerada um desafio para a veracidade dos dados. Embora o estudo seja antigo (foi realizado em 2015), o mesmo apresenta *insights* interessantes sobre a relação entre o BD e a simulação de combate, favorecendo a identificação de casos de uso para a elaboração de uma arquitetura de integração de dados.

Tabela 3.1: Propriedades relacionadas à simulação em BD (Adapt. de Song *et al.* [13]).

Casos	**Nível**	**Entidades**	**Instâncias**	**Resolução**	**Velocidade**	**Duração**
Experimentos Conjuntos	C, M, E	Milhões	Algumas	Alta	Tempo-real	Semanas
Análise de Linhas de Ação	C, M, E	Milhares	Centenas, Milhares	Baixa, Média	Mais-que-tempo-real	Horas

Legenda: C = Campanha, M = Missão, E = Engajamento.

Tabela 3.2: Propriedades relacionadas à geração de dados (Adapt. de Song *et al.* [13]).

Casos	**Dados**	**Geração**	**Escala**	**Aplicação**
Experimentos Conjuntos	dados de sensores, *status* das entidades	1ms	TB	Efetividade de sensores ISR. Resultados estatísticos da execução da missão
Análise de Linhas de Ação	comportamento civil, cultura, armamentos, terreno, mortes, vítimas e outros	<1seg	GB/TB	Análise de execução de tarefas e efetividade operacional

3.2 Ciclo de Vida da Engenharia de Dados

A integração de dados permite mover ou visualizar dados de diferentes fontes, consolidá-los com mais dados e aplicar transformações para que os dados estejam de acordo com as necessidades da organização [81]. Reis e Housley [12] afirmam que o ciclo de vida da engenharia de dados (ver Figura 3.1) compreende as fases de transformação os dados brutos em um produto final útil para consumo por analistas, cientistas de dados, engenheiros de *Machine Learning* (ML) e outros. Os autores também apresentam o conceito de *undercurrents*, que são os aspectos que perpassam todos os estágios do ciclo de vida, sendo nomeadas como *Aspectos Transversais* para fins do presente trabalho. Desta forma, os Estágios do Ciclo de Vida, assim como os Aspectos Transversais são os seguintes:

- **Geração**: são as fontes dos dados usadas durante o ciclo de vida. É necessário compreender com que frequência, velocidade e variedade os dados são gerados;
- **Armazenamento**: é onde e como os dados são armazenados. Permeia todo o ciclo de vida dos dados e impacta a forma com eles são utilizados em todos os demais estágios;
- **Ingestão**: é a forma como os dados são coletados das suas fontes e ingeridos no *data pipeline*;

- **Transformação**: é a forma como os dados são modificados de sua forma original para algo mais útil, agregando valor para os consumidores. Frequentemente, este estágio está entrelaçado com outros no ciclo de vida.;
- **Servimento**: é a utilização dos dados para propósitos práticos orientados pela lógica de negócio. É onde se consolida a extração de valor para os consumidores;
- **Aspectos Transversais**: segurança, gerenciamento dos dados, *DataOps*, arquitetura de dados, orquestração, e engenharia de software.

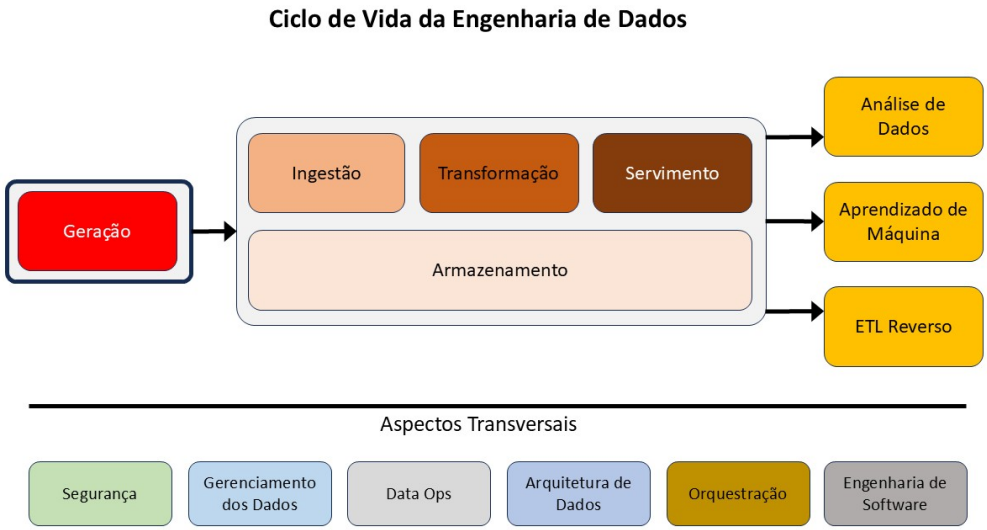

Figura 3.1: Ciclo de vida da engenharia de dados (Adaptado de Reis e Housley [12]).

Os aspectos do ciclo de vida da engenharia de dados caracterizam uma divisão didática para melhor compreensão das fases da integração de dados. Os tópicos relevantes para a proposta ora apresentada, considerando cada estágio do ciclo de vida estão dispostos nas próximas seções.

3.2.1 Geração de Dados

Reis e Housley [12] aconselham os engenheiros de dados a terem familiaridade com as fontes e como os dados são gerados. Os arquivos são um meio universal de troca de informações sendo produzidos manualmente ou como saída de outros sistemas, podendo ser estruturados, semi estruturados ou desestruturados. As APIs representam uma maneira padronizada de trocas de informações entre sistemas, simplificando a tarefa de ingestão de dados. Aplicações de bancos de dados tipo *Online Transaction Processing* (OLTP)

processam registros individuais em uma taxa elevada mas não conseguem realizar consultas que exijam examinar grandes quantidades de dados, enquanto que as aplicações do tipo *Online Analytical Processing* (OLAP) são desenhadas para processar consultas com elevada quantidade de dados, mas são ineficientes ao lidar com registros individuais. Os autores também afirmam que, tanto as aplicações tipo OLTP, quanto as tipo OLAP possuem dados passíveis de serem consultados no contexto da integração de dados. Registros de *Log* operacionais, de aplicações, de bancos de dados, de servidores, de redes e de equipamentos *Internet Of Things* (IoT) também são fontes ricas de dados a serem exploradas. Por último, mensagens relacionadas a plataformas de *streaming* são consideradas.

O contexto da simulação de combate apresenta, inicialmente, dados gerados pelos simuladores e os dados relativos ao contexto dos exercícios de simulação. No primeiro caso, a Seção 2.2 apresentou os principais padrões abertos de comunicação utilizados hoje no EB: o DIS e o *Google Protobuf*. Servidores *NodeJS* podem realizar a coleta dos dados semiestruturados gerados e armazená-los localmente até a ingestão. O DIS já possui até uma implementação inicial [82], e o software Combater possui os arquivos *.proto* necessários a implementação de um sistema de coleta de mensagens *Protobuf* [72] usando diversas bibliotecas disponíveis *online*. O tamanho aproximado de um PDU de estado da entidade (*Entity State PDU*), que é a mensagem mais frequente disseminada na rede em um exercício que usa DIS, é de cerca de 1500 *bits* [83]. Para o *Protobuf*, a documentação informa que as mensagens podem atingir até, no máximo, alguns poucos MB [84]. No segundo caso, estima-se que as informações de contexto do exercício podem possuir diversos formatos e tamanhos. Arquivos de texto, planilhas, áudio, vídeo, páginas HTML e outros formatos podem ser usados para fornecer informações a respeito da conjuntura da situação tática apresentada, sendo muito importantes para analistas e cientistas de dados.

3.2.2 Ingestão de Dados

Reis e Housley [12] apresentam considerações sobre a ingestão por lotes (*batch*) e por *streaming*, declarando que os dados são produzidos e atualizados continuadamente a partir de suas fontes. Assim, a ingestão por lotes é uma forma simples e especializada de processar a corrente de dados dividindo-a em grandes pedaços. Sob a ótica do benefício almejado, a escolha de um ou outro processo depende basicamente do caso de uso e da expectativa para acessar os dados. Os autores afirmam que ingestão de dados implica seu movimento de suas fontes para o local de armazenamento. Algumas características transpassam o processo de ingestão por lotes (*batch*) e por *streaming*, sendo importantes para o estabelecimento de uma arquitetura de integração de dados [12]:

- **Delimitação**: considerar que o fluxo de dados é dividido em intervalos de tempo, ou delimitado, para atender à contextos de domínio. No caso de ingestão por *streaming*, a preservação da natureza contínua dos dados ocorre para que possam ser processados nas etapas seguintes ainda de forma contínua;
- **Frequência**: considerar que a ingestão dos dados pode ser frequente, semi-frequente ou "em tempo real". Geralmente as frequências são misturadas dentro da organização, dependendo dos casos de uso e das tecnologias empregadas;
- **Sincronicidade**: considerar a existência de dependência e acoplamento entre a origem, a ingestão e o destino dos dados. Neste cenário, as etapas seguintes do *pipeline* não podem acontecer sem que as anteriores sejam completadas com sucesso. A ingestão assíncrona busca tornar os eventos individuais disponíveis no armazenamento assim que são individualmente ingeridos;
- **Serialização**: considerar que os dados serializados ingeridos no *pipeline* devem ser decodificados em algum momento para serem aproveitados nas etapas seguintes;
- **Atributos de Qualidade**: em relação ao desempenho e escalabilidade, considerar que o sistema deve ser desenhado para escalar e encolher usando serviços gerenciados para lidar com a vazão de dados ingeridos. Em relação à confiabilidade e durabilidade, ambos aspectos estão ligados pois alguns dados podem não ser retidos devido à falhas de ingestão;
- ***Payload***: considerar o tipo, o formato, o tamanho, o esquema e tipos de dados, e os metadados a serem ingeridos;
- **Padrões de Coleta**: considerar os padrões *push* (a origem envia dados para o destino), *pull* (o destino lê os dados da origem diretamente) e *poll* (o destino busca periodicamente por alterações nos dados da origem, lendo-os) de coleta de dados.

A decisão sobre o momento de transformar os dados também merece destaque no processo de ingestão, segundo Densmore [85]. No padrão *Extract-Transform-Load* (ETL) a transformação dos dados brutos ocorre antes deles serem armazenados em seu destino final. Já no padrão *Extract-Load-Transform* (ELT) os dados brutos são diretamente armazenados e a transformação só ocorre depois. Nos dias atuais, há uma emergência do padrão ELT devido à eficiência de leitura e escrita dos bancos de dados colunares, às tecnologias de compressão de dados, e à habilidade de distribuir processamento por múltiplos nós. O autor também afirma que, no caso da análise de dados, o ELT permite que os dados sejam armazenados sem a necessidade de se saber exatamente como eles serão utilizados, o que não era possível com o ETL. No caso de ciência de dados, o ELT também

permite a utilização direta dos dados brutos e de forma granular, atendendo os casos de uso desta área do conhecimento.

Os dados produzidos em exercícios de simulação de combate são encapsulados na duração de cada exercício. A ingestão de dados pode ocorrer usando fluxo de dados delimitado, poucas vezes por semana, não depender de outras etapas do *pipeline* e ser realizado por meio do padrão *push* para permitir maior coordenação por parte do órgão responsável. A serialização das mensagens gravadas pode ser um problema nas etapas seguintes da integração, podendo ser realizada a sua decodificação ainda antes da ingestão para evitar que os dados armazenados fiquem inertes, ou seja, sem serem utilizados ou estarem inacessíveis. Com relação ao momento da transformação, o fato de não haverem casos de uso claramente definidos para a exploração dos dados de simulação, conduz-se a utilização do padrão ELT. Assim, o processo de ingestão de dados por lotes parece ser o mais indicado para o domínio considerado.

3.2.3 Transformação de Dados

Os dados devem ser transformados de sua forma original para algo útil para os casos de uso na integração de dados, criando valor para consumo. Segundo Reis e Housley [12], as transformações mapeiam os tipos corretos de dados colocando os registros em formatos padronizados. Em estágios mais avançados, pode ocorrer mudanças no esquema de dados e aplicação de normalização. Mais à frente no *pipeline* os dados são agregados em larga escala para atender o caso de uso em *analytics*, ou categorizados para serem usados em processos de ML. Os autores explicam que transformações de dados geralmente estão entrelaçadas com outros estágios do ciclo de vida, ocorrendo nos sistemas de origem ou durante a ingestão. Neste contexto, a lógica de negócio é o principal aspecto norteador, direcionando a implementação das transformações a serem executadas, sempre de forma padronizada e automatizada.

Steen e Tanenbaum [86] apresentam a computação em *cluster* como uma das formas de computação distribuída de alto desempenho. A mesma é caracterizada por um conjunto de nós computacionais, conectados em uma rede local de alta velocidade com cada nó operando um mesmo sistema operacional. O sistema é considerado financeiramente e tecnicamente atrativo mesmo com múltiplas CPU com múltiplos núcleos, e a utilização de uma rede separada para monitorar os nós de processamento. Bani *et al.* [87] afirmam que o *Massively Parallel Processing* (MPP) ou Processamento Paralelo Massivo permite gerenciar grandes volumes de dados e prover consultas, relatórios, *dashboards* e análises, por meio da distribuição da carga de dados e processamento em diversos servidores ou clientes. Outro aspecto, Marz e Warren [88] consideram a baixa latência nas consultas e atualizações com uma das propriedades de um sistema de *Big Data*. Os autores afirmam

que a grande maioria das aplicações requer leituras com latência de até algumas centenas de milissegundos, sem comprometer a robustez do sistema.

A transformação dos dados de simulação de combate tem por objetivo extrair valor em proveito do preparo das tropas e do desenvolvimento da doutrina de emprego. Estes contextos norteiam os processos de transformação e consulta massiva permitindo a escalabilidade necessária ao domínio. Da mesma forma, a falta de casos de uso para os dados coletados também é um aspecto norteador para as operações de transformação. Para isso, as transformações devem ser operacionalizadas pela computação distribuída de alto desempenho, por processos automatizados, pela conservação dos dados coletados, e por soluções de otimização das consultas.

3.2.4 Armazenamento de Dados

A escolha da solução de armazenamento é crucial para o sucesso da arquitetura, pois permeia todos os estágios do ciclo de vida da engenharia de dados, afirmam Reis e Housley [12]. Este tópico pode ser analisado sob a ótica de seus sistemas e das abstrações empregadas. No caso dos sistemas, eles representam uma camada lógica sobre os componentes brutos de hardware, como os discos rígidos, por exemplo. O processamento de grande volume de dados torna os padrões de armazenamento e acesso complexos, demandando a distribuição dos dados entre vários servidores, formando um sistema de armazenamento distribuído que coordena o depósito, recuperação e processamentos dos dados de forma mais rápida, com maior escala e maior tolerância à falhas. Assim, os principais sistemas de armazenamento são [12]:

- **Armazenamento de Arquivos**: são sistemas que organizam arquivos em uma estrutura de diretórios e utilizam o sistema operacional para armazenar metadados sobre arquivos e diretórios, o que inclui permissões e apontadores para estas entidades;
- **Armazenamento em Blocos**: é o armazenamento bruto de blocos, que são a menor unidade que pode ser endereçável de dados em um disco, sendo fornecido por *Solid State Drive* (SSD) e discos rígidos;
- **Armazenamento de Objetos**: é o armazenamento de arquivos imutáveis de diversos tipos e tamanhos, sendo o termo objeto utilizado como um construto que representa estes arquivos;
- **Armazenamento de *Streaming***: em filas de mensagens, os dados são armazenados temporariamente. O *streaming storage* compreende a retenção destes dados por longos períodos.

As abstrações de armazenamento da engenharia de dados visam a organização dos dados e os padrões de consultas, estando no centro do ciclo de vida da engenharia de dados. São estruturas construídas sobre os sistemas de armazenamento, devendo ser analisadas com base em seu propósito, caso de uso, padrões de atualização, custos e separação entre armazenamento e processamento. O último fator é uma tendência nos dias atuais. Desta forma, as principais abstrações de armazenamento são, ainda segundo Reis e Housley [12]:

- ***Data Warehouse***: refere-se a uma plataforma tecnológica, uma arquitetura para centralização, e um padrão organizacional dentro de um empreendimento. Embora possuam capacidade de processamento massivo, não conseguem lidar com dados verdadeiramente não estruturados, como imagens, vídeos ou áudio;
- ***Data Lake***: repositório massivo de dados brutos não processado, com retenção de elevadas quantidades de dados com menor custo. Possuem grande flexibilidade e escalabilidade, além de permitirem a utilização de tecnologias *Massively Parallel Processing* (MPP), fornecendo funcionalidades avançadas no gerenciamento dos dados;
- ***Data Lakehouse***: é uma combinação entre o *Data Warehouse* e o *Data Lake*. Utiliza armazenamento de objetos ao mesmo tempo que inclui esquemas e tabelas;
- ***Data Platforms***: são ecossistemas de ferramentas interoperáveis com alta integração com a camada de armazenamento. É considerada uma abordagem não amadurecida;
- ***Stream-to-Batch***: é quando dados fluindo por um tópico de um sistema de armazenamento de *streaming* é servido para múltiplos consumidores, que podem necessitar de dados em 'tempo real' ou por lotes.

Os dados gerados por exercícios de simulação podem ser de diversos tipos. Além dos dados gerados pelos sistemas de simulação, podem ser produzidos arquivos de áudio, vídeo, planilhas e documentos de texto que descrevem atividades operacionais e comunicações realizadas, fornecendo o contexto necessário à extração de valor dos dados produzidos. Considerando as características da simulação de combate e as alternativas de armazenamento, as abstrações mais adequadas para o domínio são: *Data Lake* ou *Data Lakehouse*.

3.2.5 Servimento de Dados

Os dados são valiosos quando são usados para propósitos práticos. Entre as formas usualmente utilizadas para extrair valor dos dados estão *Analytics* e Inteligência Artificial

(IA). Em relação ao primeiro caso, Simon [89] e Skyrius [90] apresentam alguns tipos de análises que podem retirar *insights* dos dados servidos, juntamente com as perguntas que as caracterizam:

- **Descritiva**: descrevem algo que aconteceu no passado ou acontece no presente. São análises baseadas em dados históricos e dependem de sua qualidade e dos métodos aplicados (O quê aconteceu? O que está acontecendo agora?);
- **Diagnóstica**: apresentam uma visão mais profunda sobre as causas do que foi encontrado na análise descritiva (Por quê aconteceu?);
- **Preditiva**: mostram uma previsão aproximada para tendências futuras e comportamentos, sendo baseada na extração de informações de dados coletados (O quê pode acontecer?);
- **Exploratória**: aprofunda o conhecimento de uma grande quantidade de dados buscando padrões interessantes e importantes ou outros achados que podem estar escondidos (Há algo de interessante ou importante neste volume de dados?);
- **Prescritiva**: apresenta possíveis linhas de ação provenientes da análise dos dados disponíveis, sugerindo a mais vantajosa (O quê pode ser feito? Quais são as opções?).

Em relação ao segundo caso, uma das definições de IA é a da capacidade que sistemas computacionais realizam tarefas que normalmente requerem inteligência humana [91]. Russel e Norwig [92] afirmam que os principais ramos da área compreendem o processamento da linguagem natural, a representação do conhecimento, o raciocínio automatizado, o aprendizado de máquina, a visão computacional e a robótica. Alguns exemplos de trabalhos já realizados nesta área incluem: aplicações de aprendizado profundo para ensinar agentes sintéticos a operarem em ambientes de treinamento de combates aéreos [93]; implementação de métodos para capturar, caracterizar e replicar comportamentos da força oponente para tornar um sistema de simulação mais desafiador [94]; e desenvolver uma estrutura multinível para analisar os dados coletados de vídeo, áudio e *logs* de simulação para avaliar o desempenho de uma equipe em treinamento [95].

O servimento dos dados consolida a obtenção do valor extraído e a finalidade de uma integração de dados. No caso da simulação de combate, os diversos tipos de análises em dados produzidos pelos simuladores pode revelar achados importantes com relação ao preparo da tropa e à informações doutrinárias usadas em operações militares. Da mesma forma, a aplicação da IA sobre estes mesmos dados pode proporcionar desde o desenvolvimento de modelos de Aprendizagem de Máquina para automatizar tarefas, até modelos autônomos que possam auxiliar na tomada de decisão ou fazer o papel do inimigo.

3.2.6 Aspectos Transversais

Os aspectos transversais representam o movimento de elevação da visão arquitetural do ciclo de vida da engenharia de dados, permeando todos os seus estágios, como visto na Seção 3.2. Dentre estas particularidades foram destacadas algumas que têm maior importância para o trabalho apresentado, pois são consideradas essenciais para a implantação inicial de uma arquitetura de integração de dados. Assim, Reis e Housley [12] tecem as seguintes considerações sobre estas características:

- **Segurança**: deve ser enfatizada a cultura de segurança em toda a organização. A regra é exercitar o princípio do mínimo privilégio, o que significa que os usuários só podem acessar recursos essenciais para executar a função pretendida;
- **Governança**: é a capacidade de assegurar a qualidade, integridade, segurança e usabilidade dos dados coletados. A qualidade é a conformidade dos dados com a expectativa do negócio, e possui três características: precisão, completude e oportunidade. Os dados têm que estar disponíveis para acesso rápido e confiável. Os metadados são exatamente aqueles dados que permitem a descoberta e a governabilidade de toda a estrutura. Por fim, a linhagem de dados, caracterizada por um processo de registro de uma trilha de auditagem dos dados durante o ciclo de vida;
- **Orquestração**: observabilidade, monitoramento, registro, alerta e rastreamento são críticos para estar à frente de problemas no ciclo de vida da engenharia de dados. Um componente importante neste contexto é o orquestrador, ou motor que coordena as muitas tarefas o mais rápido e eficientemente possível, em uma cadência planejada. Um sistema orquestrador permanece *online* em elevada disponibilidade, o que permite o monitoramento constante de sistemas e ferramentas, e condições de erro, enviando alertas oportunamente;
- ***Open Source***: é um modelo de distribuição de software e do código subjacente, tornando-o disponível para uso geral sob termos de licenciamento específicos. Eles são classificados em: gerenciados pela comunidade e comerciais;
- ***Containers***: frequentemente referidos como máquinas virtuais leves, isolam espaços de utilização de um único sistema operacional. Apresentam alguns dos principais benefícios da virtualização, como gerenciamento de dependências e isolamento de código.

A implantação de uma arquitetura de integração de dados deve considerar muitos dos aspectos transversais. A governança de dados e a segurança são assuntos que merecem destaque quando analisados sob a ótica da simulação de combate. O primeiro porque está

ligado diretamente ao aproveitamento dos dados coletados, possibilitando o êxito na sua extração de valor e evitando o aparecimento do *Data Swamp*, ou uma coleção bagunçada com conjuntos de dados difíceis de encontrar, e tempo de resposta muito baixo [96]. A segunda pelo fato de que a atividade militar possui um elevado grau inerente de sigilo, e as informações utilizadas e produzidas por simuladores estão abrangidas dentro deste contexto. Os demais aspectos transversais são consideradas como meios técnicos para atingir os objetivos de negócio.

3.3 Atributos de Qualidade de Software

Um proposta para integração de dados de simulação de combate deve ser orientada por visão abrangente da arquitetura. Richards e Ford [97] afirmam que, embora a arquitetura de software se refira a um mapa ou planta para o desenvolvimento de um sistema, não é possível definir suas partes de forma precisa, revelando a dificuldade de formular uma expressão do todo. Eles definem quatro dimensões constituintes de uma arquitetura: estrutura, características arquiteturais, decisões arquiteturais e princípios de *design*. As dimensões se referem ao tipo de estilo arquitetural a ser implementado, os critérios de sucesso do sistema, a regras sobre como o sistema deve ser construído e a o direcionamento geral do *design*. Bass *et al.* [98], afirmam que a arquitetura de software de um sistema é um conjunto de estruturas necessárias para compreendê-lo, sendo o mesmo composto de elementos de software, das relações entre eles e das propriedades de ambos. Estes autores também apresentam a arquitetura como um conjunto de estruturas geralmente muito complexas de entender quando apresentadas por completo.

Cervantes e Kazman [99] apresentam o conceito de Vetores Arquiteturais (*Architecture Drivers*) que compreendem as decisões que transformam o propósito de *design*, as funcionalidades primárias, os atributos de qualidade, as restrições, e as preocupações arquiteturais em estruturas usadas para orientar o projeto. O **propósito do *design*** consolida as razões pelas quais se adota determinado desenho e quais objetivos de negócio são mais importantes para a organização no momento. As **funcionalidades primárias** são geralmente definidas como aquelas que são críticas para atingir os objetivos de negócio, assim como aquelas que tenha elevado grau de dificuldade técnica ou que necessite da interação de muitos elementos arquiteturais. Os **atributos de qualidade** são definidos como propriedades mensuráveis e testáveis de um sistema que são utilizadas para verificar se satisfaz as necessidades de suas partes interessadas. Entre os vetores arquiteturais, são aqueles que mais influenciam no desenho da arquitetura, devendo ser adequadamente elicitados, especificados, priorizados e validados. As **restrições** se apresentam da forma de tecnologias obrigatórias, sistemas externos os quais exista necessidade de interoperabi-

lidade, leis e padrões de conformidade, retrocompatibilidade, entre outros fatores os quais se tem nenhuma ou pouca capacidade de controle. As **preocupações arquiteturais** representam aspectos adicionais considerados durante o desenho da arquitetura mas que não são expressas especificamente em documentos de requisitos.

Os atributos de qualidade, como mencionado, têm grande importância da formulação da arquitetura. Desta forma, o conhecimento destes atributos e suas características são fundamentais para o desenvolvimento da proposta apresentada nesta dissertação. De acordo com Richards e Ford [100] e com a Norma ISO/IEC 25010 [101], as definições relativas aos atributos de qualidade são as seguintes:

- **Desempenho**: medida de eficiência em relação à quantidade de recursos utilizados sob condições determinadas. Inclui comportamento de tempo (medida de resposta, tempo de processamento, e taxa de transferência), utilização de recursos (quantidades e tipos de recursos utilizados), e capacidade (em que grau os limites máximos de processamento são excedidos);
- **Segurança**: grau de proteção do software em relação às suas informações e dados, de forma que pessoas e sistemas tenham o grau apropriado de acesso coerente com seus tipos e níveis de autorização. Inclui confidencialidade (os dados somente são acessíveis a quem tem autorização), integridade (o sistema não permite acesso não autorizado, modificação de seu código ou dados), auditabilidade (rastreabilidade de ações ou eventos), e autenticidade (acreditação da identidade de um usuário);
- **Escalabilidade**: habilidade do sistema de atingir boa performance e operar a medida que o número de usuários ou requisições aumenta;
- **Confiabilidade**: grau de funcionamento do sistema sob condições específicas por um período de tempo específico. Inclui maturidade (confiabilidade sob condições normais de operação), tolerância a falhas (operação esperada mesmo com falhas de software e hardware), e recuperação (capacidade de se recuperar dados após uma falha e restabelecer seu estado);
- **Manutenibilidade**: grau de eficácia ou eficiência com que desenvolvedores podem modificar o software no intuito de melhorar, corrigir, ou adaptar às mudanças no ambiente e requisitos. Inclui modularidade (constituição do software em componentes discretos), reusabilidade (reutilização de um componente em mais de um sistema ou na construção de outros componentes), modificabilidade (modificação de um software sem degradar a qualidade do produto) e testabilidade (facilidade em testar o software;

- **Disponibilidade**: tempo durante o qual o sistema está disponível e tempo para reestabelecimento no caso de ocorrência de uma falha;
- **Compatibilidade**: grau de capacidade de troca de informações de um sistema ou componente com outros sistemas e componentes, e de executar suas funcionalidades enquanto compartilha o mesmo hardware ou ambiente de software. Inclui coexistência (compartilhamento de ambiente comum e recursos com outros produtos) e interoperabilidade (grau de capacidade de troca de informações entre dois ou mais sistemas);
- **Usabilidade**: efetividade, eficiência e satisfação dos usuários com o sistema em relação ao seu propósito. Inclui reconhecimento de apropriabilidade do software, facilidade de aprendizagem pelos usuários, proteção de erros de uso, e acessibilidade mais ampla possível.

3.4 Trabalhos Relacionados

O estudo de trabalhos relacionados buscou compreender os aspectos arquiteturais da integração de dados de saúde, assim como identificar as características das *frameworks* de simulação. No primeiro caso, os artigos foram avaliados com ênfase nos aspectos relacionados ao ciclo de vida de engenharia de dados, tais como: geração, ingestão, transformação, armazenamento, e servimento. No segundo caso, os atributos visados tinham por finalidade estabelecer analogias entre os *frameworks* e a estrutura de coleta e processamento de dados como fontes de conhecimento experimental, ao mesmo tempo que deveriam fornecer apoio para a realização dos treinamentos militares.

3.4.1 Arquiteturas de Referência

A integração de dados médicos possui uma quantidade considerável de trabalhos publicados, o que parece estar ligado a uma maior demanda de conhecimento sobre o assunto. Kraus *et al.* [102] trabalharam com a medicina de precisão, uma abordagem médica que leva em consideração as características individuais do paciente, como algo que simboliza o progresso na terapêutica por transigir de tratamentos baseados em sintomas para terapias orientadas em marcadores genéticos e moleculares. Ele reconhece que os dados clínicos frequentemente não estão acessíveis devido a fatores como a heterogeneidade das diversas fontes envolvidas, e acredita que o *big data* tem o potencial de melhorar os tratamentos de saúde em termos de medicina de precisão, modelagem preditiva, apoio à decisões clínicas e pesquisa científica comparativa. Porém, antes disso, há a necessidade de resolver questões básicas a respeito da disponibilidade de dados.

A escolha do estudo deste tipo de integração de dados encontra motivo em fatores quantitativos e qualitativos. A demanda por conhecimento deste assunto levou a uma maior quantidade de artigos produzidos o que induz, empiricamente, haver uma certa consolidação de conhecimentos e experiências quanto as arquiteturas estudadas. Desta forma, alguns dos artigos disponíveis foram selecionados levando em conta o horizonte temporal de cinco anos, a relevância do assunto abordado e a semelhança com o objeto de estudo deste trabalho. Para fins do apresentado neste estudo, Centros de Integração de Dados (CID) é designação genérica do local para onde os dados são destinados após a ingestão, considerando o contexto da arquitetura de integração de dados.

Prasser *et al.* [103] propôs conceitos e soluções nos níveis técnico e organizacional, tendo foco específico na integração e no compartilhamento de dados. O contexto do trabalho é o consórcio de Integração de Dados para a Medicina do Futuro (DIFUTURE, na sigla em inglês), que pretende estabelecer CID nos centros médicos universitários na Alemanha. A proposta apresentou a operação dos CID como unidades encarregadas da integração de dados médicos de vários tipos, oriundos de fontes clínicas e de pesquisa, em uma abordagem de três fases. Na primeira, os dados são importados e harmonizados, usando padrões da indústria para dados e interfaces. Na segunda, os dados são pré-processados, transformados, harmonizados e enriquecidos dentro de áreas lógicas de armazenamento. Na terceira e última fase, os dados transformados são importados em plataformas de *analytics* e modelagem de dados, sendo disponibilizados em formatos compatíveis com requisitos de interoperabilidade definidos previamente. A segurança no acesso e no compartilhamento são apresentadas por meio da combinação entre tecnologias de melhoria de privacidade e métodos de computação distribuída.

Winter *et al.* [104] apresentaram resultados sobre os problemas arquiteturais no *design* de integração de dados da Tecnologia de Informação Médica Inteligente para Saúde (SMITH na sigla em inglês), um consórcio alemão composto de universidades, hospitais universitários, instituições de pesquisa e companhias de TI. SMITH utiliza uma abordagem federada para sua estrutura de governança e um desenho de um conceito genérico para seus CID, que compartilham funcionalidades e serviços com o intuito de melhor aproveitar as arquiteturas de interoperabilidade e de uso e acesso à dados. O CID provê acesso aos registros médicos eletrônicos dos hospitais locais com base em serviços de acreditação de dados e privacidade. O compartilhamento de dados clínicos e de pesquisa é baseado em padrões de comunicação e armazenamento. A arquitetura de referência dos CID determina os serviços, aplicações e as ligações de comunicação baseada em padrões, provendo ingestão, enriquecimento, transformação, e tarefas de transferência de dados.

Stufi *et al.* [105] demonstraram a implementação de uma plataforma de *Big Data Analytics* (BDA) para atender aos requisitos impostos pelo Serviço de Saúde Nacional da

República Tcheca. A plataforma suporta capacidades analíticas e algoritmos de Inteligência Artificial (IA) por meio de *Machine Learning* (ML) e mineração de dados. Uma prova de conceito foi desenvolvida e elevada a ambiente de produção com a intenção de unificar todas as partes isoladas do sistema de saúde em um ecossistema de integração de dados. A plataforma é caracterizada por ser um sistema distribuído de grande escala, composto de três componentes-chave: as soluções *Talend* [106] (análise *ad hoc*, gerenciamento de dados e metadados, e integração de dados), *Vertica* [107] (armazenamento e processamento de dados) e *Tableau* [108] (visualização de dados).

Wang *et al.* [109] desenvolveram uma plataforma de BDA de saúde em um grande hospital da China, tendo a governança de dados como conceito central. A finalidade é resolver dificuldades relacionadas à integração, processamento, armazenamento, padronização e segurança de dados heterogêneos de múltiplas fontes, gerados em mais de cem departamentos e seções da instituição. A plataforma integra todos os sistemas de operação do hospital para gerar dados de alta qualidade e formar um banco de dados massivo multidimensional que possa apoiar, de forma abrangente, atividades de clínica médica, pesquisa científica e gestão hospitalar. A integração de dados é obtida por meio da sincronização de dados armazenados em diversos locais com um banco de dados mestre, e os arquivos muito grandes são enviados via *File Transfer Protocol* (FTP) para locais de armazenamento de objetos.

Parciak *et al.* [110] implementaram um *framework* automatizado para processamento oportuno de dados clínicos em um hospital universitário da Alemanha. O trabalho demonstrou a aplicação da proposta descrevendo sua utilização em um CID por meio de um protótipo, caracterizado por ser baseado em microsserviços e totalmente *open-source*. A estrutura de automatização de processamento de dados incorpora o registro completo das atividades de gerenciamento e manipulação dos dados, e inclui um esquema de metadados para rastreabilidade da proveniência deles, e um conceito de validação do processo. Os principais requisitos do CID são: entrada de dados de muitas fontes heterogêneas, pseudonimização e harmonização de dados, integração em um *data warehouse*, e possibilidade de extração ou agregação de dados para pesquisa científica de acordo com requisitos de proteção. Os dados brutos são armazenados em um *data lake*, transferidos para um *data warehouse* relacional e separadas em subconjuntos, estes são novamente transformados e armazenados para consumo atendendo à casos de uso para pesquisas científicas.

Hoffmann *et al.* [111] propuseram o desenvolvimento de um *framework* genérico que integra dados de saúde e *analytics* em um software clínico que suporta tanto decisões relativas à prática clínica quanto os esforços de pesquisa médica. O trabalho alemão desenvolveu uma aplicação *web* que integra análise de dados, visualização, assim como simulações computacionais e modelos preditivos. Dados de pacientes de fontes variadas,

descentralizadas e heterogêneas são extraídos, transformados e carregados em um CID que funciona como um repositório de dados clínicos. Uma aplicação de integração e análise de dados conectada a um servidor de simulação é utilizada em pesquisas médicas e na tomada de decisão de tratamentos de pacientes.

A análise dos artigos selecionados revelou os aspectos comuns presentes em todas as arquiteturas de integração de dados na área de saúde estudadas. Em primeiro lugar, todos os trabalhos apresentaram utilização de processamento massivo e distribuído na fase de transformação dos dados, assim como objetivavam transformar esses dados para sua utilização em *analytics* e Inteligência Artificial (IA). Em segundo lugar, ao avaliar os aspectos relativos ao armazenamento, atestou-se a utilização de mecanismos de gerenciamento de dados e metadados, e de rastreabilidade e controle de qualidade dos dados. Em terceiro lugar, quanto ao servimento dos dados, observou-se que se dava por meio de plataforma própria, tudo com vistas a atender uso operacional e em pesquisa. Por fim, foi verificado o uso de protocolos de interoperabilidade nas comunicações dentro do *pipeline* e controle de acesso por meio de processos de autenticação.

Os aspectos que apresentaram discrepâncias entre as arquiteturas apresentadas nos estudos estão resumidas na Tabela 3.3. A maioria dos estudos demonstrou arquiteturas que coletavam dados de múltiplas fontes distribuídas geograficamente, e relatou a utilização da coleta de dados por *batch* e usando ELT durante a fase de ingestão de dados. Em relação ao armazenamento, a maioria dos estudos relatou utilizar bancos de dados distribuídos e mecanismos de catalogação dos dados. Quanto ao servimento, a maioria dos autores relatou a utilização de ferramentas de visualização dos dados. Em relação às soluções utilizadas, a maioria das arquiteturas propostas utiliza mecanismos de monitoramento e orquestração, tecnologias *open source* e ambientes virtualizados usando *containers*. Pouco mais da metade dos trabalhos replica os dados armazenados e apenas três trabalhos mencionam que tem suas redes segregadas da *Imternet*.

A comparação apresentada ajuda a inferir que a arquitetura de integração de dados de saúde, em sua visão mais holística, apresenta grandes similaridades. Ao observar as discrepâncias entre os estudos, verifica-se que aspectos relativos à ingestão e coleta de dados e soluções empregadas são os pontos de maior divergência, o que ocorre pela peculiaridade de cada caso de uso. Por exemplo, Wang *et al.* [109] utilizou uma abordagem de sincronização de banco de dados tendo por base o paradigma mestre-escravo, o que levou a uma ingestão direta de dados oriundos do banco de dados, e a inexistência de mecanismos de orquestração. Assim, pode-se inferir que os aspectos mais gerais da arquitetura de integração de dados estão representados nos estudos analisados e as particularidades de cada estudo podem ser relevadas, ou até descartadas, por representarem aspectos específicos de cada caso de uso. Em outras palavras, a amostra de estudos representa a maioria

Tabela 3.3: Características das arquiteturas de referência.

Característica da Arquitetura	[103]	[104]	[105]	[109]	[110]	[111]	**T**
Distribuição geográfica das fontes	✔	✔	✔	✖	✖	✔	✔
Coleta de dados por lotes (*batch*)	✔	-	✔	✔	✔	✔	✔
Coleta de dados por meio de ELT	✔	-	✔	✖	✖	✖	✔
Armazenamento fisicamente distribuído	✔	✔	✔	✔	✔	-	✔
Replicação dos dados armazenados	✔	✔	✖	✖	✔	-	✔
Catalogação dos dados armazenados	✔	✔	✔	✔	✔	-	✔
Monitoramento e orquestração	✔	-	✔	✖	✔	✔	✔
Ferramentas de visualização	✔	-	✔	✔	✔	✔	✔
Segregação da *Internet*	-	✖	✔	-	✔	✖	✔
Tecnologias *open source*	✔	-	✔	✔	✔	✔	✔
Virtualização usando *containers*	✔	✔	✔	-	✔	-	✔

Legenda: ✔= Presente, ✖= Ausente, **-** = Informação indisponível, **T** = Este Trabalho.

das características da arquitetura referência de integração de dados, objeto de estudo do presente trabalho.

3.4.2 *Frameworks* de Simulação

Os *frameworks* de simulação representam a estrutura de código subjacente responsável pela simulação em si. Lu *et al.* [112] explicam que o motor de simulação é responsável pelo controle do tempo, da simulação e do armazenamento, enquanto que o *framework* provê um conjunto de ferramentas de código para o desenvolvimento de um sistema de simulação que, frequentemente, inclui um motor de simulação ou tem elevado acoplamento com este. Estes *frameworks* têm a capacidade de executar a simulação em um ambiente distribuído, de ser decomposto em partes menores com interfaces padronizadas, de escalar positiva ou negativamente com facilidade, e de ser aplicada em outros cenários, de forma genérica.

A generalização proporcionada pelos *frameworks* de simulação em comparação aos sistemas de simulação, como aqueles vistos na Seção 2.4, proporciona a utilização destas estruturas de código em proveito da experimentação. King *et al.* [113] afirmam que as múltiplas representações de um mesmo modelo conceitual em diferentes fidelidades, resoluções e detalhes são ideais para o processo de experimentação, assim *frameworks* fornecem a infraestrutura e os meios necessários para criar e montar modelos específicos que representam o sistema de interesse para um determinado propósito. Assim, as características encontradas nas funcionalidades destas estruturas de código podem ser reaproveitadas em outros contextos, como o da experimentação de defesa.

Clive *et al.* [114] e West e Birkmire [115], apresentaram artigos descrevendo o *Advanced Framework for Simulation, Integration and Modeling* (AFSIM) desenvolvido pela empresa *Boeing* e, atualmente, sob controle da *United States Air Force* (USAF). É uma estrutura de simulação multidomínio e multirresolução, utilizada para análise, experimentação e treinamento, cuja a finalidade original era o desenvolvimento de novos conceitos de sistemas para a Força Aérea Americana. O *framework* provê uma arquitetura e serviços que permitem a criação e execução de cenários de simulação de forma construtiva ou virtual, apoiada por um conjunto de bibliotecas de software, uma interface gráfica para visualização, e um conjunto de entidades de simulação. Estas entidades e seus subsistemas podem ser modelados e reutilizados, e representam plataformas de combate de terra, ar, mar e espaço. O AFSIM teve destacado papel no desenvolvimento, amadurecimento e teste de algoritmos para veículos aéreos autônomos.

Dantas *et al.* [116] [117], demonstraram o Ambiente de Simulação Aeroespacial (ASA) que permite simulação de cenários operacionais para apoiar o desenvolvimento de táticas e procedimentos no contexto aeroespacial, jogos de guerra, avaliação de equipamentos e desenvolvimento de tecnologias, tudo em proveito da Força Aérea Brasileira (FAB). Possui em sua arquitetura componentes de desenvolvimento e execução de cenários, ferramentas de ciência de dados e visualização. Os resultados das simulações são armazenados em um banco de dados dedicado e integrados a uma plataforma colaborativa. Entre as atividades de pesquisa e desenvolvimento que utilizam o ASA estão o desenvolvimento de algoritmos de IA com variados propósitos, como por exemplo aplicações de sistemas de combate e tráfego aéreo. É importante ressaltar que o ASA tem o AFSIM como modelo e referência, e que o *framework* foi desenvolvido usando o motor de simulação *Mixed Reality Simulation Platform* (MIXR) [118], um software criado para aplicações de simulação robustas, escaláveis, virtuais, construtivas, *stand-alone* e distribuídas.

O *framework Flexible Analysis and Mission Effectiveness System* (FLAMES) é uma família de softwares *Commercial Off-The-Shelf* (COTS) para desenvolvimento de simulações construtivas e virtuais com interface *Live, Virtual and Constructive* (LVC). Ele provê

uma infraestrutura que é independente de qualquer simulador específico, fornecendo ferramentas para edição e execução de cenários, para desenvolvimento de ferramentas e outros. Possui uma arquitetura *plug-and-play* na qual os *plugins* (componentes) são integrados à aplicação FLAMES, sendo os responsáveis pelos aspectos específicos à cada simulação. O *framework* também dispõe de integração com o motor de jogos digitais *Unreal* [119] o que estende suas funcionalidades permitindo a criação de *serious games* e simulações construtivas e virtuais, com grande riqueza visual [120].

Os *frameworks* estudados, resumidos na Tabela 3.4, apresentaram algumas semelhanças e diferenças. Todos eles podem realizar simulação construtiva e virtual de forma nativa, possuem compatibilidade com os padrões de integração de simuladores como o DIS e o HLA, têm funcionalidades para geração e coleta de dados das simulações, permitem uso operacional (em treinamento) e em pesquisa (experimentação), além de possuírem ferramentas de visualização de dados. Com relação às diferenças entre eles, observa-se em alguns a impossibilidade de utilização de plataforma *web* como interface com o usuário, assim como a inexistência de um ambiente colaborativo de ciência de dados. Nenhum deles possui capacidade de simulação viva de forma nativa. Não é possível concluir sobre as discrepâncias relativas ao processamento de dados massivo e distribuído, e à capacidade de transformação e armazenamento de dados.

Tabela 3.4: Características dos *frameworks* de simulação.

Característica da *Framework*	[116][117]	[114][115]	[120]	**T**
Emprego da simulação viva	✗	✗	✗	✓
Processamento de dados massivo e distribuído	✓	-	✓	✓
Capacidade de transformação e armazenamento	✓	-	-	✓
Plataforma *web* para gerenciamento	✓	✗	✗	✓
Uso operacional/treinamento	✓	✓	✓	✓
Uso em pesquisa/experimentação	✓	✓	✓	✓
Ambiente colaborativo para ciência de dados	✓	-	✗	✓
Ferramentas de visualização	✓	✓	✓	✓

Legenda: ✓= Presente, ✗= Ausente, **-** = Informação indisponível, **T** = Este Trabalho.

As características observadas revelam uma nova possibilidade de utilização de um *pipeline* de dados, uma vez que o EB não possui um *framework* de simulação. Desta forma, uma infraestrutura de integração de dados pode atender tanto ao treinamento quanto a experimentação de defesa. Os *frameworks* estudados estão entre os mais empregados em modelagem e simulação, segundo Lu *et al.* [112] e King *et al.*, [113] o que fortalece o

argumento apresentado. Outro aspecto observado foi de que tanto o AFSIM, e o ASA possuem comunidade de usuários de organizações parceiras, sendo que o AFSIM possui até um modelo de licenciamento de compartilhamento de informações com representantes da indústria e da academia. Tal situação pode representar um rol maior de beneficiários de uma estrutura de integração de dados de simulação combate, sendo composta não apenas de usuários militares (governo), mas também pelos demais representantes da tríplice hélice. O contexto apresentado pela tese da tríplice hélice apresenta o governo, a indústria e a academia trabalhando de forma sinérgica para gerar inovação e desenvolvimento [121] e, no caso apresentado, aproveitam-se da infraestrutura de simulação existente.

O estudo das arquiteturas de integração de dados de saúde e dos *frameworks* de simulação forneceram os subsídios para a elaboração de uma proposta para uma arquitetura de integração de dados de simulação de combate no âmbito da F Ter. As características encontradas foram identificadas com **(*C-n*)** e agrupadas, considerando as fases do ciclo de vida de engenharia de dados, como visto na Seção 3.2, favorecendo a tomada de decisões de *design*:

- **Geração**: múltiplas fontes (sistemas de simulação) distribuídas geograficamente, com dados originados de simulação virtual, construtiva e viva **(*C-1*)**;
- **Ingestão**: coleta de dados por lotes **(*C-2*)** e ELT **(*C-3*)**;
- **Transformação**: processamento de dados massivo e distribuído, e transformação de dados para fins de *Analytics* **(*C-4*)** e IA **(*C-5*)**;
- **Armazenamento**: armazenamento fisicamente distribuído e gerenciamento dos metadados **(*C-6*)**, da replicação, rastreabilidade, qualidade e catalogação dos dados **(*C-7*)**;
- **Servimento**: dados servidos por meio de plataforma (*web*), para uso operacional e em pesquisa **(*C-8*)**, existência de ferramentas de visualização de dados **(*C-9*)** e de ambiente colaborativo de ciência de dados **(*C-10*)**;
- **Aspectos Transversais**: compatibilidade com padrões de integração de simuladores **(*C-11*)**, monitoramento e orquestração do *pipeline* de dados **(*C-12*)**, acesso controlado ao ambiente **(*C-13*)**, rede segregada da *Internet* **(*C-14*)**, uso de tecnologias *open source* **(*C-15*)**, e virtualização usando *containers* **(*C-16*)**.

3.5 Conclusões Parciais

Neste capítulo foram apresentados alguns aspectos que influenciam a forma como uma arquitetura de integração de dados para simulação de combate pode ser concebida. As

características do *big data*, em especial a variedade, volume e velocidade dos dados gerados pelos simuladores apresentam características peculiares se comparadas com outros domínios. Na geração de dados, verifica-se a necessidade de coletá-los das simulações juntamente com arquivos que apresentam o contexto tático dos exercícios, sob pena de torná-los inertes. Na ingestão de dados, atesta-se que a coleta por lotes usando ELT parece ser o processo mais adequado para o ingresso no *data pipeline* pela inexistência de casos de uso definidos para extração de valor dos dados. A padronização e automatização de processos de transformação orientados para os objetivos de negócio (treinamento e doutrina), aliado ao paradigma do *Data Lake* ou *Data Lakehouse*, dão robustez e flexibilidade ao tratamento dos vários tipos de dados produzidos. O servimento orientado ao consumo para *analytics* e IA materializa a obtenção de valor, objetivo final da arquitetura. Os aspectos transversais, em particular a governança de dados e a segurança, devem ser observados durante todos os estágios do ciclo de vida da engenharia de dados, favorecendo o sucesso da integração almejada. Ainda nesta direção, os atributos de qualidade de software fornecem as métricas necessárias para validar o *pipeline* de dados, certificando o atingimento dos objetivos de negócio. Finalmente, o estudos de trabalhos relacionados sobre arquiteturas de referência e *frameworks* de simulação extraiu as melhores práticas de integração de dados e de utilização de simulações em proveito do treinamento e doutrina, revelando as características almejadas em uma solução de integração de dados de simulação de combate. Desta forma, foram reunidos os conhecimentos necessários para a elaboração da proposta deste trabalho, apresentada no próximo capítulo.

Capítulo 4

Arquitetura Proposta

O presente capítulo tem por objetivo demonstrar os processos de implementação e validação da arquitetura, sendo apresentado em três seções. A Seção 4.1 apresenta o desenho da solução em seu mais alto nível. A Seção 4.2 detalha a forma com as funcionalidades principais são implementadas, alinhadas às decisões arquiteturais previamente tomadas. A Seção 4.3 busca validar a arquitetura por meio de um experimento controlado e da análise de métricas previamente definidas.

Os capítulos anteriores apresentaram as peculiaridades do emprego da simulação de combate, os conceitos de engenharia de dados, e trabalhos relacionados aderentes à proposta deste trabalho. O conhecimento de domínio proporcionado pelos estudos realizados forneceu subsídios para a definição de características as quais são desejáveis na solução proposta. A partir deste ponto, é necessário o estabelecimento de um desenho arquitetural consistente e alinhado com as melhores práticas encontradas. Em seguida, deve ser realizada a implementação de um protótipo de escopo limitado mas suficiente para ser submetido a um experimento prático de validação. Por fim, a análise dos achados no experimento pode atestar a viabilidade de uma arquitetura de integração de dados de simulação de combate, o que materializa a solução da questão central do trabalho.

4.1 Desenho do Protótipo

O desenho do protótipo é caracterizado pela visualização inicial do desenvolvimento da arquitetura. Inicialmente, foi elaborada uma concepção do modelo de negócio para facilitar a compreensão dos aspectos de domínio de integração de dados de simulação de combate. A seguir, foram elicitados requisitos de alto nível tendo como base o modelo de negócio e as características levantadas no estudo das arquiteturas de referência e dos *frameworks* de simulação (Seção 3.4). Por último, foi realizada a análise de componentes e seus relacionamentos, tudo embasado nas decisões arquiteturais específicas para garantir

aderência ao problema em questão. As seções seguintes apresentam cada uma das etapas mencionadas, pormenorizando e justificando as decisões de concepção do protótipo.

4.1.1 Modelo de Negócio

A simulação de combate no EB é essencial para o preparo da F Ter e a experimentação doutrinária, sendo aplicada no treinamento de tropas, especialmente das FORPRON, e no teste de hipóteses para o desenvolvimento de capacidades militares. Nesse contexto, os CA e CI desempenham papéis cruciais ao manipular os meios de simulação e fornecer pessoal especializado. A coleta de dados dos exercícios é fundamental para avaliar o desempenho e embasar análises doutrinárias, sendo realizada de forma agendada, já que os exercícios são atividades pontuais no tempo e no espaço, permitindo análise a posteriori sem necessidade de monitoramento em tempo real. Os sistemas de simulação do EB abrangem as categorias virtual, viva e construtiva, com compatibilidade com protocolos abertos, como o *Google Protocol Buffers*, possibilitando o desenvolvimento de interfaces customizadas para integração e coleta de dados.

Esses sistemas estão distribuídos pelo território nacional, exigindo uma arquitetura que integre os CA e CI. O contexto do problema demonstra a necessidade de extração de valor de dados produzidos no escopo do preparo da F Ter e da experimentação de defesa, ambos apoiados pela simulação de combate, como facilitadores da missão constitucional do EB em defender a Pátria, como demonstrado ao longo do Capítulo 2 (ver Seções 2.3 e 2.4).

Assim sendo, o modelo de negócio vislumbrado para a arquitetura de integração de dados de simulação de combate tem por objetivo extrair valor dos dados gerados pelos sistemas de simulação. Como visto anteriormente, a F Ter se utiliza da simulação para apoiar o treinamento militar de suas tropas e o desenvolvimento de sua doutrina operacional. Os dados gerados pelos simuladores podem ser coletados, armazenados, transformados e consumidos em proveito de ambos os contextos.

A infraestrutura de integração de dados baseada na Intranet do EB, ou *EBNet*, pode proporcionar o desenvolvimento de modelos de IA para utilização no contexto da simulação de combate, apoiar análises estatísticas e experimentação doutrinária das tropas em treinamento, permitir a elaboração de relatórios periódicos ou *ad-hoc* sobre adestramentos da FORPRON ou até monitorar os próprios simuladores quanto ao seu regime de utilização. *Analytics*, IA e ferramentas de visualização são os principais meios para a multiplicar a relevância destes dados, o que pode potencializar a geração de força do EB (ver Figura 4.1).

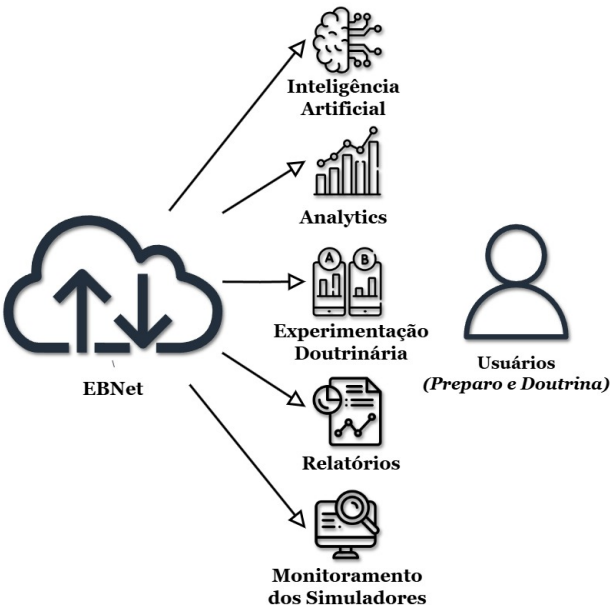

Figura 4.1: Modelo de negócio.

4.1.2 Requisitos de Alto Nível

O desenvolvimento de requisitos de alto nível encontra amparo nas características das arquiteturas de referência e das *frameworks* de simulação (ver a Seção 3.4), sendo alinhados ao modelo de negócio. Da mesma forma, os vetores arquiteturais (ver a Seção 3.3) servem como base metodológica para a elicitação destes requisitos que, concernentes com os demais aspectos de domínio, favorecem a consolidação da visão inicial da proposta.

O propósito do *design*, ou estado final desejado do sistema, é implementar um arquitetura inicial de integração de dados abrangendo todas as fases do ciclo de vida de engenharia de dados de simulação, usando a infraestrutura existente na *EBNet*. Para isso, desenvolvemos um protótipo de um *data pipeline* que consome dados gerados sinteticamente, com testes realizados no ambiente controlado de uma rede local. A implementação do protótipo tem dois objetivos: validar a arquitetura e levantar valores relativos às características de *big data* dos dados gerados pelos simuladores.

As funcionalidades primárias necessárias para o alcance dos objetivos de negócio foram identificadas com base nos achados da Seção 3.4, os quais serviram de fundamento para a elaboração da lista de requisitos funcionais consolidada na Tabela 4.1. Já os requisitos não funcionais, também extraídos das características observadas, representam restrições (*constraints*) e estão reunidos na Tabela 4.2. As preocupações arquiteturais (*architectural concerns*) pertinentes à proposta estão descritas na Tabela 4.3, destacando-se o receio quanto ao mero armazenamento de dados e à possível inércia do sistema, conforme discutido na Seção 3.2.6.

Por fim, foi realizada uma elicitação inicial dos atributos de qualidade desejáveis em

uma arquitetura de integração de dados. Esses atributos estão apresentados na Tabela 4.4 e são posteriormente quantificados e qualificados na fase de validação, mantendo alinhamento com os demais vetores arquiteturais.

Tabela 4.1: Requisitos funcionais.

Requisito Funcional (RF)	**Descrição**	**Característica Associada**
RF-1	O sistema deve ser capaz de coletar os dados dos simuladores e armazená-los localmente até o momento do ingresso dos mesmos no *pipeline*.	C-1, C-11
RF-2	O sistema deve realizar a movimentação periódica dos dados dos armazenamentos locais até o local de armazenamento centralizado.	C-2, C-3
RF-3	O sistema deve ser capaz de transformar os dados para utilização em *analytics*.	C-4
RF-4	O sistema deve ser capaz de transformar os dados para utilização em ciência de dados (IA).	C-5
RF-5	O sistema deve armazenar e gerenciar os metadados produzidos.	C-6
RF-6	O sistema deve armazenar os dados e gerenciar sua replicação, rastreabilidade, qualidade e catalogação.	C-7
RF-7	O sistema deve permitir o acesso aos dados de treinamento (uso operacional) e experimentação (pesquisa) por meio de plataforma *web*.	C-8
RF-8	O sistema deve possuir ferramentas de visualização de dados.	C-9
RF-9	O sistema deve possuir ambiente colaborativo de ciência de dados.	C-10
RF-10	O sistema deve monitorar o estado de seus componentes e coordenar as movimentações de dados.	C-1, C-2, C-3 e C-12

Tabela 4.2: Requisitos não-funcionais.

Requisito Não-Funcional (RNF)	**Descrição**	**Característica Associada**
RNF-1	O sistema deve possuir controles de acesso.	C-13
RNF-2	O sistema deve rodar em rede segregada da *Internet*.	C-14
RNF-3	Os componentes do sistema devem ser desenvolvidos em código aberto (tecnologia *open source*).	C-15
RNF-4	Os componentes do sistema devem ser executados por meio de *containers* virtualizados.	C-16

Tabela 4.3: Preocupações arquiteturais (*Arquitectural Concerns*).

Preocupação Arquitetural (PRE)	**Descrição**	**Característica Associada**
PRE-1: *Data Swamp*	Todos os aspectos da implementação e da governança de dados devem facilitar a utilização dos dados armazenados, evitando o abandono no *Data Lake*.	C-4

4.1.3 Componentes e Relacionamentos

A arquitetura, em seu mais alto nível, é dividida em camadas e estas em componentes com responsabilidades definidas. Steen e Tanenbaum [122] afirmam que a arquitetura em camadas organiza componentes em uma hierarquia, na qual cada camada oferece serviços à camada superior e consome serviços da camada inferior, o que promove modularidade e abstração. Consideram uma estrutura eficaz em sistemas distribuídos onde interfaces bem definidas ocultam detalhes de implementação, garantindo a separação de responsabilidades e facilitando a manutenção. Por outro lado, os autores apresentam a arquitetura orientada a serviços que organiza sistemas como uma composição de serviços independentes, cada um encapsulando uma funcionalidade específica por meio de interfaces padronizadas. Tais serviços podem ser executados em processos ou máquinas distintas, o que promove a interoperabilidade e a reusabilidade.

A arquitetura proposta combina os dois estilos arquiteturais, estruturando o sistema em camadas lógicas que encapsulam serviços especializados. As camadas são desenhadas para cumprir funções lógicas específicas no *pipeline* de dados, ao mesmo tempo que os componentes encerram serviços que garantem flexibilidade e interoperabilidade. A ideia é aliar a estrutura clara da lógica de processamento das camadas com o desacoplamento e portabilidade dos serviços componentes. Essa abordagem híbrida atende às necessidades de negócio para integrar dados de simulação de combate, garantindo escalabilidade, flexibilidade e eficiência, ao mesmo tempo que mitiga desafios associados à heterogeneidade e ao volume de dados.

A estrutura proposta é descrita com auxílio da Figura 4.2, e o foco inicial é coletar os dados das aplicações de simulação de combate, transformá-los e servi-los aos consumidores, que são os responsáveis pelo preparo da tropa e pela formulação da doutrina militar. A arquitetura é caracterizada pelas camadas de gerenciamento do sistema, de orquestração e monitoramento, de ingestão, de transformação e de servimento. Desta forma, apresenta coerência com os princípios do ciclo de vida da engenharia de dados (ver Seção 3.2). Todos os componentes devem ser oriundos de projetos desenvolvidos em código aberto e executados por meio de *containers* virtualizados, tudo visando atender os requisitos

Tabela 4.4: Atributos de Qualidade (AQ).

ID	**Atributo**	**Métrica**	**Requisito Associado**
AQ-1	Desempenho	O sistema deve ser capaz de coletar e armazenar eventos locais com elevada eficácia.	RF-1
AQ-2	Desempenho	O sistema deve ser capaz de mover os dados dos armazenamentos locais para o *data lake* com baixa latência.	RF-2
AQ-3	Desempenho	O sistema deve ser capaz de mover os dados dos armazenamentos com elevada eficiência.	RF-2
AQ-4	Desempenho	O sistema deve ser capaz transformar dados para *analytics* e ciência de dados com elevada eficiência.	RF-3, RF-4
AQ-5	Desempenho	O sistema deve ser realizar consultas de dados para *analytics* e ciência de dados com elevada eficiência.	RF-3, RF-4
AQ-6	Confiabilidade	Os dados armazenados devem possuir elevada consistência.	RF-5, RF-6
AQ-7	Confiabilidade	Os dados armazenados devem possuir elevada rastreabilidade.	RF-5, RF-6
AQ-8	Desempenho	O sistema deve disponibilizar o funcionamento da plataforma *web* com reduzida latência.	RF-7, RF-8, RF-9
AQ-9	Desempenho	O sistema deve ser capaz de monitorar seu estado com elevada eficiência.	RF-10
AQ-10	Usabilidade	O sistema deve permitir acesso simples e intuitivo aos serviços da plataforma.	Todos
AQ-11	Disponibilidade	O sistema deve permanecer constantemente disponível e operante.	Todos
AQ-12	Escalabilidade	O sistema deve possuir a capacidade de receber mais pontos de integração.	Todos
AQ-13	Segurança	O sistema deve operar sem incidentes de segurança.	Todos
AQ-14	Confiabilidade	O sistema deve recuperar-se de falhas de operação rapidamente.	Todos
AQ-15	Confiabilidade	O sistema deve operar com máxima estabilidade.	Todos
AQ-16	Manutenibilidade	O sistema deve operar com capacidade de substituição de componentes.	Todos

não-funcionais RNF-3 e RNF-4.

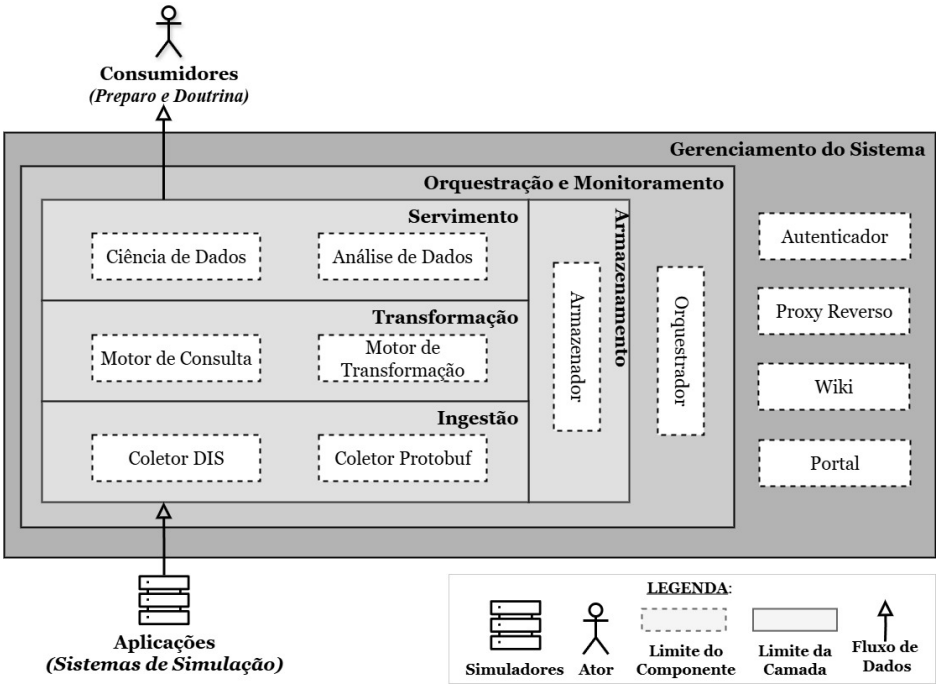

Figura 4.2: Estrutura proposta.

Gerenciamento do Sistema

O sistema como um todo deve operar dentro da intranet do EB, atendendo ao requisito não-funcional RNF-2. A camada de Gerenciamento do Sistema é composta pelos serviços responsáveis pelo acesso à estrutura de integração de dados de simulação de combate. A solução proposta atende aos Aspectos Transversais, em especial, da Governança e da Segurança proporcionada pelos componentes da arquitetura (ver a Seção 3.2.6). Os componentes se relacionam com as camadas mais internas da arquitetura por meio do redirecionamento controlado de requisições de usuários e do fluxo de mensagens de sistemas. Os componentes da camada estão descritos a seguir:

- **Autenticador**: é o componente que provê autenticação e autorização para usuários e serviços do sistema. Tem por finalidade atender o requisito não-funcional RNF-1;
- ***Proxy* Reverso**: um *proxy* reverso é um componente intermediário que traduz requisições externas a uma rede e as encaminha para servidores dentro da rede, geralmente usado para encapsular serviços de informação [123]. O componente tem a função de encapsular a arquitetura, dando maior segurança para as comunicações internas entre os componentes e controlando as requisições que saem dela. O componente atende ao requisito não-funcional RNF-1;

- **Wiki**: é o componente responsável pela base de conhecimentos do sistema, facilitando a descoberta e organização dos dados, mitigando a preocupação arquitetural PRE-1;
- **Portal**: é a fachada do sistema para os usuários, contendo todos os serviços disponíveis de acordo com os perfis e níveis de acesso correspondentes. Visa atender o requisito funcional RF-7.

Orquestração e Monitoramento

A camada de orquestração e monitoramento é o cérebro da arquitetura, gerenciando a movimentação dos dados e observando o estado dos serviços em operação. Possui um único componente chamado **Orquestrador** que organiza e automatiza os complexos fluxos de tarefas agendadas, e as dependências entre elas, além de garantir o caminho crítico de execução. Também monitora as métricas de qualidade dos serviços. Tudo se coaduna com o mencionado na Seção 3.2.6. O componente tem por objetivo atender aos requisitos funcionais RF-2 e RF-10.

Ingestão

A camada de ingestão é responsável pela facilitação do ingresso dos dados gerados pelos sistemas de simulação. Possui dois componentes especializados que têm por finalidade capturar e armazenar as mensagens locais das redes montadas para apoiar os exercícios de simulação, desta forma atendendo ao requisito funcional RF-1. A ingestão deve seguir o padrão ELT, com *push* dos dados para o repositório, coleta de dados por lotes, e independência com relação às etapas seguintes do *pipeline*, tudo conforme mencionado na Seção 3.2.2. Estes componentes devem coletar mensagens DIS e *Protobuf* que representam a totalidade dos protocolos de comunicação de simuladores dentro do escopo da proposta (ver Seção 2.4), conforme apresentado a seguir:

- **Coletor DIS**: componente de coleta de dados no protocolo de comunicação no padrão DIS, conforme descrito na Seção 2.2.3; e
- **Coletor *Protobuf***: componente de coleta de dados no protocolo de comunicação *Protobuf* do software MASA Combater, conforme visto na Seção 2.4.2.

Armazenamento

A camada de armazenamento é o coração do sistema, proporcionando a persistência dos dados coletados e facilitando a sua transformação e utilização. Esta camada tem um único componente especializado chamado **Armazenador**, no qual arquivos de diversos

formatos devem ser depositados após serem ingeridos pelo *pipeline*. Podem ser armazenados arquivos de áudio, vídeo, planilhas, e documentos de texto, além dos arquivos relacionados à coleta de dados de simuladores (ver Seção 3.2.4). O componente atende aos requisitos funcionais RF-5 e RF-6.

Transformação

A camada de transformação tem por finalidade intermediar as transformações de dados e otimizar as consultas aos dados armazenados. A inexistência de casos de uso definidos levam a uma abordagem flexível na qual a computação distribuída, processos automatizados, conservação de dados (em combinação com o armazenamento) são características importantes no contexto de domínio (ver Seção 3.2.3). Assim, a camada de transformação tem dois componentes com a finalidade de preparar os dados para seu uso em *analytics* e IA, o que atende aos requisitos funcionais RF-4 e RF-5. Os componentes desta camada são os seguintes:

- **Motor de Transformação**: tem por objetivo realizar a transformação de dados em larga escala por meio de processamento massivo e distribuído, para fins de preparar os dados para consumo;
- **Motor de Consulta**: é um componente intermediário entre as bases de dados transformadas e as ferramentas de consumo da arquitetura, que tem por finalidade otimizar o desempenho das consultas.

Servimento

A camada de servimento é responsável por permitir que os usuários do sistema possam explorar e extrair valor dos dados coletados. Proporciona a realização de análises de dados ou a aplicação da IA nas aplicações militares em proveito do preparo da F Ter, e do desenvolvimento da Doutrina Militar Terrestre (ver Seção 3.2.5), caracterizando o objetivo final da arquitetura. Esta camada é composta por um componente de ciência de dados e outro para análise de dados, conforme descrito a seguir:

- **Ciência de Dados**: este componente é um ambiente colaborativo que utiliza *notebooks jupyter* para acessar os dados armazenados. O objetivo é permitir a exploração dos dados e o desenvolvimento de análises estatísticas, mineração de dados, aprendizado profundo e desenvolvimento de modelos de IA. O componente atende aos requisitos RF-7 e RF-9;
- **Análise de Dados**: é um serviço de acesso aos dados da arquitetura através de relatórios *ad-hoc* ou de *dashbords* para visualização de informações com a finalidade

de acompanhar processos correntes da organização. Este componente atende aos requisitos RF-7 e RF-8.

4.2 Implementação do Protótipo

O desenvolvimento do protótipo envolveu a escolha das tecnologias utilizadas nos componentes da arquitetura e a compreensão de como esses componentes foram implementados. Para garantir consistência nas decisões arquiteturais, foi essencial fundamentar as escolhas feitas em relação às soluções adotadas. Assim, os aspectos de implementação apresentados correspondem ao detalhamento das principais etapas do desenvolvimento, com o objetivo de descrever de forma mais abrangente o processo de construção do software.

As tecnologias utilizadas no protótipo implementado, estão alinhadas com as decisões arquiteturais previamente tomadas. A Seção 4.1.2 elencou os requisitos de alto nível da arquitetura que são compostos por Requisitos Funcionais (RF), Requisitos Não-Funcionais (RNF), Preocupações Arquiteturais (PRE), e Atributos de Qualidade desejáveis. Desta forma, cada componente deve cooperar para o atendimento dos requisitos de alto nível, possuindo as funcionalidades requeridas nos RF, restringindo-se aos RNF, mitigando as PRE, e favorecendo que a arquitetura possua os atributos de qualidade almejados, tudo conforme visto na Seção 4.1.2. A seguir as seções apresentam as tecnologias utilizadas, com respectiva fundamentação para sua escolha, e os principais aspectos de implementação.

4.2.1 Autenticador

O componente de autenticação é o **Keycloak** [124] que é uma solução de gestão de autenticação e autorização de acesso. É compatível com o protocolo de autenticação *OpenID Connect* [125] que permite a interoperabilidade entre os serviços, simplificando a identificação de usuários e sistemas por meio de um servidor de autorização. Esta característica permite o que se chama de *Single Sign-on* (SSO), que é a capacidade do usuário poder se autenticar uma única vez para todas as aplicações da arquitetura [126].

Além disso, o *Keycloak* possui interoperabilidade com os demais serviços da arquitetura que são expostos externamente aos usuários (ver Tabela 4.5), o que favorece sua utilização como uma solução de segurança no sistema. Possui código aberto, tendo sido utilizada a imagem *Docker* da versão 26.0.0 disponível publicamente [127]. Desta forma, a tecnologia escolhida atende aos requisitos RNF-1, RNF-3, RNF-4, além de propiciar usabilidade (AQ-8) e segurança (AQ-11).

Tabela 4.5: Interoperabilidade do *Keycloak*.

Solução	**Componente**	**Referências**
Nginx	Proxy Reverso	[128]
Wiki.js	Portal/Wiki	[129], [130]
Airflow	Orquestrador	[131]
MinIO	Armazenador	[132], [133]
Dremio	Motor de Consulta	[134]
JupyterHub	Ciência de Dados	[135]
Superset	Análise de Dados	[136]

4.2.2 Proxy Reverso

O **Nginx** [137] que é um software que pode ser empregado como servidor *web*, **proxy** reverso, balanceador de carga, e outras funções. No contexto da implementação, ele redireciona as requisições externas, sejam de usuários ou de simuladores, para dentro da arquitetura. Internamente, os serviços se comunicam diretamente entre si usando o protocolo *HyperText Transfer Protocol* (HTTP). O *Nginx* é um software de código aberto, cuja versão empregada foi a 1.28, com imagem *Docker* [138] disponível. Com isso, ele atende aos requisitos RNF-1, RNF-3, RNF-4, além de cooperar com a segurança (AQ-11).

4.2.3 Wiki

A *wiki* do sistema é o **Wiki.js** [139] que é uma solução colaborativa para compartilhamento e organização de informações corporativas. É um software de código aberto e que possui imagem *Docker* pública [140]. A tecnologia facilita o atendimento dos princípios FAIR (*Findable*, *Acessible*, *Interoperable*, *Reusable*) [141], buscando tornar os dados da arquitetura encontráveis, acessíveis, interoperáveis e reutilizáveis atendendo tanto dados abertos quanto de acesso restrito, tudo por meio da base de conhecimentos gerenciada pela *wiki*.

A solução é enxertada no Portal e organizada nas seções: *Home*, Planejamento, Execução, e Documentação (da arquitetura), havendo uma ligação entre as páginas *web* e os dados oriundos das simulações, fornecendo contexto a estes. A versão utilizada foi a 2.5, de código aberto e com imagem *Docker* pública [140]. A escolha do *Wiki.js* atende aos requisitos RF-6, RNF-3, RNF-4, mitiga a PRE-1, bem como favorece a usabilidade (AQ-8).

4.2.4 Portal

Para este componente, foi realizada a implementação customizada de dois serviços, tendo o **Node.js** [142] como tecnologia empregada. O primeiro serviço é a própria fachada do sistema apresentando todos os serviços disponíveis, conforme o perfil do usuário autenticado e suas permissões, sendo caracterizada por um servidor simples de página *web* que redireciona para os serviços adequados ao nível de autorização do usuário. O segundo serviço é uma aplicação de criação e gerenciamento de exercícios com a finalidade de fornecer contexto tático aos dados coletados a partir dos simuladores, conforme discutido na Seção 3.2.4. A sua finalidade é coordenar a associação entre arquivos de simulação oriundos dos coletores de mensagens DIS e *Protobuf*, e os arquivos de contexto tático disponíveis. A cada criação de exercício, um arquivo ***exercicio-metadata.json*** é gerado para rastrear os respectivos metadados (ver Seção 4.2.7).

O gerenciamento dos exercícios é realizado por meio de um arquivo ***manifesto.json*** (ver Seção 4.2.7) que contém referências a todas as pastas de exercícios do sistema e seus arquivos de metadados. Uma vez que há a vinculação entre os dados de contexto tático e os arquivos de simulações de um exercício, a pasta correspondente é movimentada para o local onde está pronta para a transformação. A Figura 4.3 apresenta um diagrama de sequência para a criação dos exercícios.

A tecnologia usada é de código aberto, e a versão utilizada foi a 23.0.0, com imagem *Docker* [143] disponível. Desta forma, o componente está coerente com os requisitos RF-7, RNF-3, RNF-4, assim como beneficia o desempenho (AQ-7) e usabilidade (AQ-8) da arquitetura.

4.2.5 Orquestrador

O **Apache Airflow** [144] foi a solução usada na implementação do orquestrador. É uma plataforma de código aberto para desenvolvimento, agendamento e monitoramento de fluxos de tarefas por lotes. Ele possui uma *framework Python* que permite criar fluxos de tarefas que se conectam com praticamente qualquer tecnologia, além de uma interface *web* para monitoramento. Isso é possível graças à utilização da estrutura de código chamada *Directed Acyclic Graph* (DAG), que é uma representação baseada em grafo na qual as tarefas são os vértices, e as dependências entre elas são arestas unidirecionais. A DAG permite a execução de tarefas em paralelo, e o rastreamento e monitoramento de seus resultados [145].

A implementação com componente emprega duas DAGs para fazer o processamento da transformação dos dados ingeridos pela arquitetura, assim como coletar suas métricas. A primeira identifica os exercícios prontos para processamento, determina a execução do

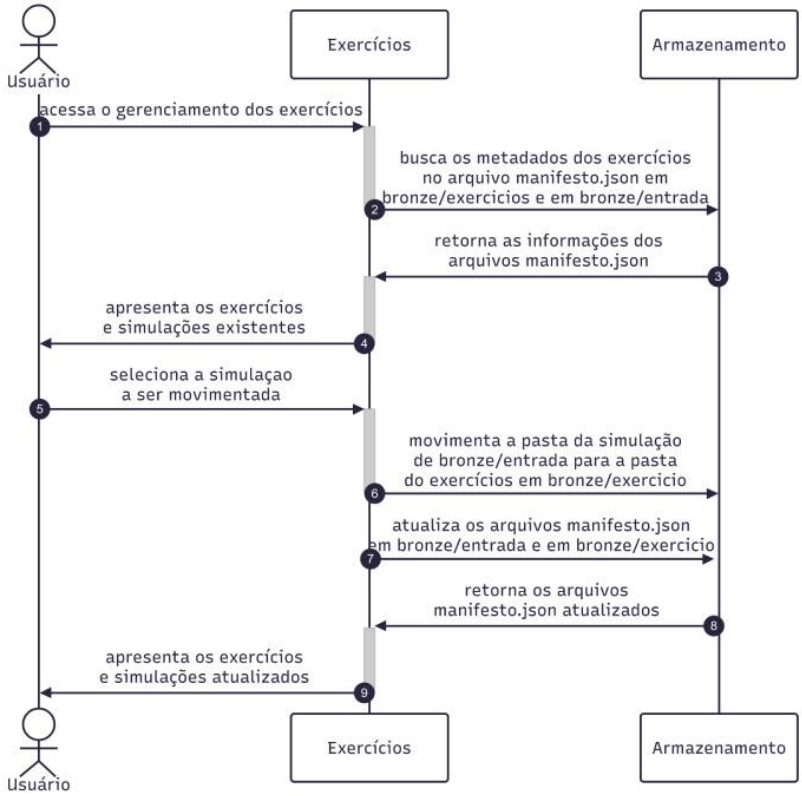

Figura 4.3: Diagrama de sequência de criação dos exercícios.

Job Spark de transformação e realiza o armazenamento dos dados (simulação e contexto) no formato *Iceberg*, e no final atualiza os metadados (ver Seção 4.2.8) do **Armazenador**. A segunda coleta as métricas do *Nginx*, do *MinIO*, do *Spark*, do *Dremio*, e do próprio *Airflow*. Isso é realizado por meio de conexões internas da arquitetura, entre o **Orquestrador** e os respectivos serviços, além de leituras de arquivos de *log* localizados em pasta compartilhada (no caso do *Nginx*). Neste contexto, é importante informar que os serviços *Wiki.js*, *Keycloak*, *Airflow*, *Superset*, *Project Nessie*, e as próprias métricas da arquitetura empregam o banco de dados relacional *PostgreSQL* [146], o que facilita a interoperabilidade na coleta das métricas.

A solução é de código aberto e apresenta uma imagem *Docker* [147] oficial da versão 2.11.0, implementada no protótipo da arquitetura. Pelas características apresentadas, a tecnologia atende aos requisitos RF-2, RF-10, RNF-3, RNF-4, além de cooperar com o desempenho (AQ-3) do sistema.

4.2.6 Coletores

Neste componente foram desenvolvidos dois serviços especializados na coleta de dados de simulação de combate usando **Node.js** [142], um para o protocolo DIS e outro para o

Protocol Buffers. O primeiro foi desenvolvido usando a biblioteca *Open-DIS* [27], tendo a implementação *Open DIS for Javascript* [82] sido usada como ponto de partida da solução desenvolvida. No segundo caso, o desenvolvimento se utilizou da biblioteca *Protobuf.js* [148] juntamente com conjuntos de arquivos *.proto* usados no software Combater.

Os dois tipos de coletores utilizam *WebSockets* para transmitir os dados para dentro da arquitetura. Isso promove uma comunicação segura em duas vias entre as fontes de dados (simuladores) e os coletores [149]. A ideia é facilitar a coordenação entre o início e o término do envio de mensagens aos coletores. No caso do protocolo DIS, foi necessário implementar um programa **Ouvinte** para coletar as mensagens da rede local (LAN), conforme visto na Seção 2.2.3, e retransmiti-las para seu coletor na arquitetura. Os coletores armazenam os dados localmente e fazem o *upload* dos arquivos para a pasta **Entrada** no *bucket* **Bronze** do **Armazenador**, o que ocorre tão logo todos os dados do exercício sejam recebidos. Nesta situação, os arquivos de simulação aguardam a sua associação com os arquivos de contexto tático, sendo controlados por meio do arquivo ***manifesto.json*** (ver a Seção 4.2.7) que mantém uma lista daquelas pastas que ainda precisam ser associadas. A Figura 4.4 apresenta um diagrama de sequência que ilustra o processo de coleta de dados dos simuladores.

Os coletores também foram desenvolvidos usando a imagem *Node.js* da versão 23.0.0 [143]. Desta forma, os componentes desenvolvidos atendem aos requisitos RF-1, RNF-3, RNF-4, e beneficiam o desempenho (AQ-1) da arquitetura.

4.2.7 Armazenador

O componente é um armazenador centralizado de objetos (do inglês: *object storage*), no qual os arquivos de diversos tipos são depositados após serem ingeridos pelo *pipeline* e mantidos em seu formato original. Para isso a solução escolhida é o **MinIO** [150], um armazenador de objetos que reúne todas as características de um *data lake* moderno. Ele possui elevado desempenho e escalabilidade, resiliência e capacidade de interoperabilidade por meio de conteinerização, orquestração, automação e compatibilidade com diversas APIs.

O *Armazenador* foi configurado para possuir dois *buckets*: **Bronze** e ***Warehouse***. O primeiro possui duas partições: **Entrada** e **Exercícios**. Os dados de simulação ingressam na partição de **Entrada** (ver Seção 4.2.6), sendo movidos para **Exercícios** após passarem pela associação entre dados de simuladores e dados de contexto tático, o que sinaliza que estão prontos para passar pelo processo de transformação (ver Seção 4.2.4). Após transformados, os dados são armazenados no *bucket* **Warehouse** no formato *Iceberg*. Os dados originais são mantidos e podem ser reprocessados em qualquer momento no futuro. As Figuras 4.3, 4.4, e 4.7 apresentam diagramas de sequência que ilustram os

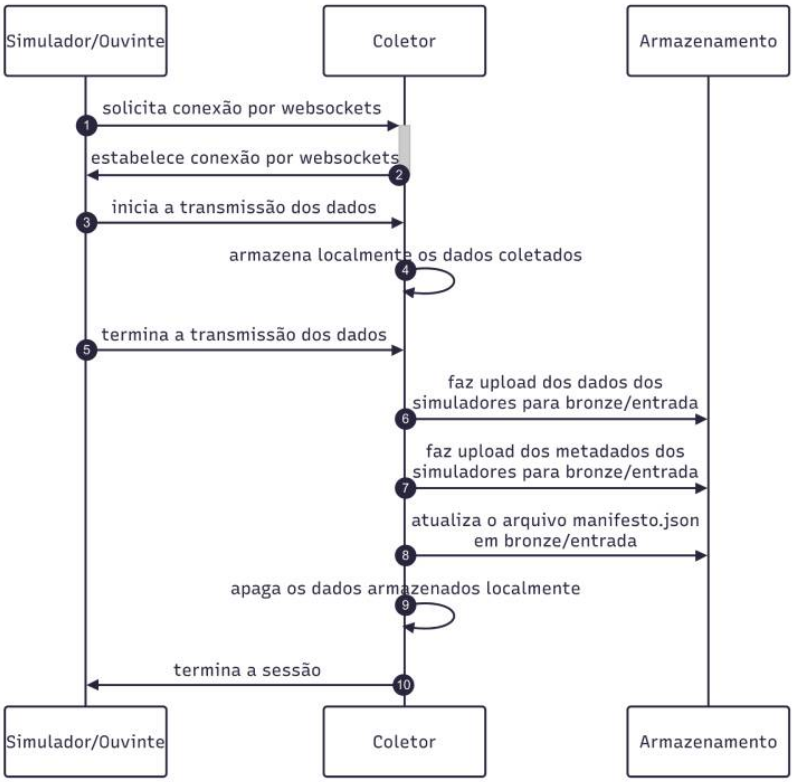

Figura 4.4: Diagrama de sequência da coleta dos dados.

fluxos de trabalho que envolvem armazenamento de objetos da arquitetura. Estes fluxos correspondem à criação de exercícios, à coleta de dados e à transformação dos dados coletados.

O *bucket* **Bronze** apresenta a estrutura ilustrada na Figura 4.5. A partição **Entrada** apresenta diversas pastas nomeadas conforme o IP e o dia da coleta de dados de simuladores. Cada pasta possui outras pastas nomeadas por hora de coleta, que representa cada sessão, sendo organizadas desta forma para facilitar o restabelecimento em caso de queda da rede. Nestas pastas estão arquivos binários de simuladores e um arquivo chamado ***sessao-metadata.json*** que apresenta os metadados de cada sessão, os quais tem sua estrutura descrita na Lista 4.1. Já a partição **Exercicios**, apresenta um conjunto de pastas que contém documentos de contexto tático e simulações associadas.

Listing 4.1: Estrutura do arquivo ***sessao-metadata.json***.

```
{
    "inicioSessao": "data de inicio",
    "terminoSessao": "data de termino",
    "duracaoEmMinutos": "duracao em minutos",
    "porta": "numero da porta",
    "mensagens": quantidade de mensagens,
    "totalKB": total em KB,
    "conexoes": quantidade de conexoes,
    "origens": [
        "ip da origem",
        ...
    ],
    "pastaPai": "ip-data",
    "pastaFilho": "timestamp",
    "arquivos": [
        "nome do arquivo",
        ...
    ],
    "arquivoMetadados": "sessao-metadata.json",
    "diretorioBase": "./dados",
    "diretorio": "./dados/ip-data/timestamp"
}
```

Cada exercício apresenta um arquivo chamado ***exercicio-metadata.json*** que guarda metadados relevantes para rastreamento, o que pode ser verificado na Lista 4.2. Na raiz de ambas as partições, arquivos ***manifesto.json*** monitoram listas de referências para pastas de arquivos de simulação (ver Lista 4.3) e de exercícios completos com arquivos de contexto tático e de simuladores (ver Lista 4.4).

Listing 4.2: Estrutura do arquivo ***exercicio-metadata.json***.

```
{
    "id": "origem-tipo-descr-data",
      "name": "nome do exercicio",
      "description": "decricao do exercicio",
      "simulationType": "viva" | "virtual" | "construtiva",
      "origin": "CA-SUL" | "CA-LESTE",
      "startDate": "data de inicio",
      "endDate": "data de termino",
      "trainedForce": "forca adestrada",
      "wikiUrl": "url do exercicio",
      "documentos": [
        {
          "id": "uuid",
          "name": "nome do documento",
          "description": "descricao do documento",
          "path": "exercises/id/documents/documento",
          "size": tamanho em bytes,
          "mimetype": "mimetype",
          "uploadedAt": "2025-05-31T14:41:21.907Z"
        }
        ...
      ],
      "simulacoes": [
        {
          "id": "ip da origem_data",
          "name": "ip da origem_data",
          "path": "exercicios/id/simulacoes/ip da origem_data",
          "originalPath": "entrada/ip da origem_data",
          "movedAt": "data da movimentacao"
        },
        ...
      ],
      "criadoEm": "data de criacao",
      "atualizadoEm": "data de atualizacao"
}
```

O *bucket* **Warehouse** é destinado ao armazenamento de arquivos pós-transformação (ver Seção 4.2.8). Nele, cada pasta de exercício possui duas outras pastas: *documentos* e *simulações*. Em *documentos*, diversos arquivos do tipo *json* contêm os dados textuais

Figura 4.5: Estrutura do *bucket* **Bronze** do **Armazenador (MinIO)**.

extraídos dos arquivos de contexto tático. Na pasta *simulacoes* estão os arquivos no formato *Iceberg*, que compreendem dados e metadados. Arquivos de dados são armazenados no formato *Apache Parquet* [151] e arquivos de metadados são salvos nos formatos *json* e *Apache Avro* [152], tudo conforme a especificação do formato.

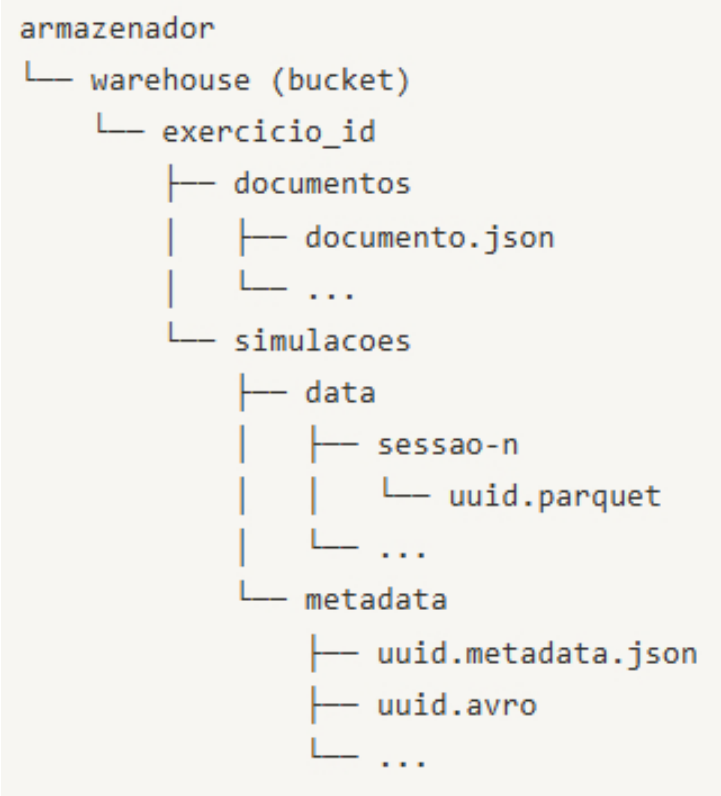

Figura 4.6: Estrutura do *bucket* **Warehouse** do **Armazenador (*MinIO*)**.

Listing 4.3: Estrutura do arquivo ***manifesto.json*** na pasta **Entrada** do *bucket* **Bronze**.

```
[
    {
        "id": "ip da origem_data da coleta",
        "dados": {
            "origem": "ip da origem",
            "caminho": "entrada/ip_data",
            "criadoEm": "data de criacao"
        }
    },
    ...
]
```

O *MinIO* é um software de código aberto e possui imagem *Docker* publicada [153], sendo usada a versão lançada em abril de 2025 (RELEASE.2025-04-22T22-12-26Z). Por essas razões, ele coopera com os requisitos RF-5, RNF-3, e RNF-4, assim como contribui para a confiabilidade (AQ-5 e AQ-6) do sistema.

Listing 4.4: Estrutura do arquivo ***manifesto.json*** na pasta **Exercícios** do *bucket* **Bronze**.

```
{
    "criados": [
        {
            "id": "origem-tipo-descr-dia",
            "dados": {
                "origem": "CA-SUL" | "CA-LESTE",
                "caminho": "exercicios/origem-tipo-descr-dia",
                "criadoEm": "data de criacao",
                "atualizadoEm": "data de atualizacao"
            },
        }
        ...
    ],
    "processados": [
            {
                "id": "origem-tipo-descr-dia",
                "dados": {
                    "origem": "CA-SUL" | "CA-LESTE",
                    "caminho": "exercicios/origem-tipo-descr-dia",
                    "criadoEm": "data de criacao",
                    "atualizadoEm": "data de atualizacao"
                    }
            },
        ...
    ]
}
```

4.2.8 Motor de Transformação

A tecnologia escolhida foi o **Apache Spark** [154], caracterizado por um motor computacional e um conjunto de bibliotecas de código para processamento paralelo em *clusters* de computadores [155]. O seu objetivo principal é transformar os dados armazenados por meio do uso do processamento massivo paralelo, facilitando a extração de valor nos contextos de *analytics* e IA. A transformação dos dados é feita sob a coordenação do **Orquestrador**, que inspeciona o arquivo ***manifesto.json*** localizado na partição **Exercícios** do *bucket* **Bronze**, com a finalidade de identificar os exercícios que estão prontos para processamento.

A seguir o *Job Spark* realiza as transformações, extraindo o texto dos arquivos .pdf de contexto e os arquivos das sessões de simulação conforme o protocolo utilizado. No caso do DIS, o *Spark* conta com o auxílio da biblioteca *Open-DIS* [27], desta vez por meio da implementação na linguagem *python* [156]. No caso do *Protobuf* foram utilizados *scripts python* gerados [157] a partir de esquemas existentes no software Combater. Após a tranformação, os dados são salvos no formato *Iceberg* no *bucket* **Warehouse** (ver a Figura 4.6).

Por fim, o **Orquestrador** atualiza o manifesto contendo as pastas com os exercícios. A Figura 4.7 descreve a sequência do processo de transformação dos dados coletados. O *Spark* possui imagem *Docker* [158] disponível, o que o torna coerente com os requisitos RF-3, RF-4, RNF-3, RNF-4, e contribuinte para o desempenho (AQ-4) da arquitetura.

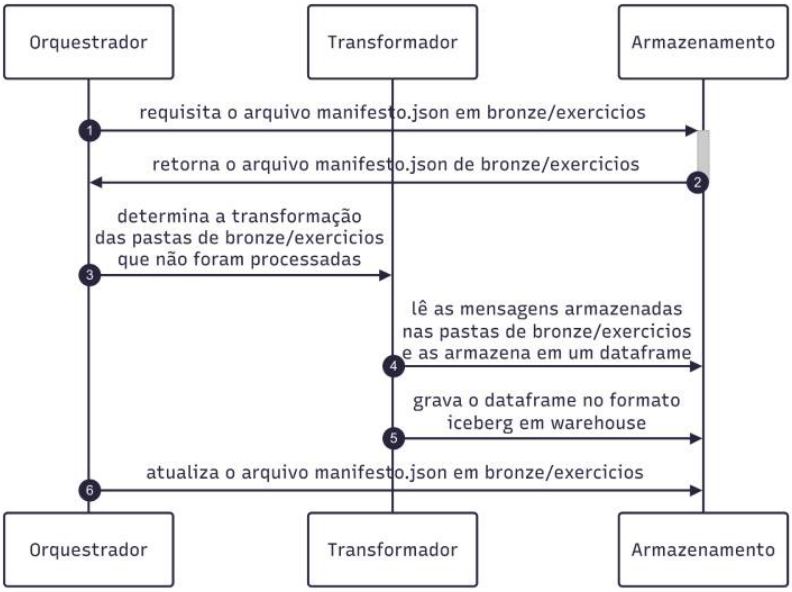

Figura 4.7: Diagrama de sequência de transformação dos dados coletados.

4.2.9 Motor de Consulta

O **Dremio** [159] é o componente principal, sendo utilizado em conjunto com o **Project Nessie** [160] e o formato de armazenamento de dados **Apache Iceberg** [161]. O *Dremio* permite consultas rápidas e unificadas em múltiplas fontes de dados sem necessidade de movimentação destes. Ele possui um mecanismo vetorizado que permite consultas em elevada velocidade, mesmo em grandes *datasets* [162]. O catálogo *Nessie* monitora as alterações nos conjuntos de dados armazenados mantendo a sua consistência. Possui funcionalidades análogas às de sistemas de controle de versão *Git*, com criação de *branches* e *commits* que permitem testes e validação de dados em ambientes isolados do repositório

principal. Isso proporciona integridade e rastreabilidade das operações sobre os dados armazenados [163].

O *Apache Iceberg* é um formato aberto de tabelas que se aproveita da computação distribuída, do armazenamento de objetos e das funcionalidades *Git* dos catálogos, fornecendo caraterísticas que excedem as funcionalidades dos *data warehouses* tradicionais. Necessita de um motor de processamento, de um armazenador de objetos escalável, e de um catálogo para rastrear todos os metadados envolvidos. Nestas condições, facilita a implementação das seguintes capacidades: evolução do esquema, particionamento oculto, evolução da organização das partições, consulta a dados históricos e retorno a versões anteriores dos dados [164].

As consultas são facilitadas pelas conexões internas da arquitetura. No caso do protótipo implementado, o *Dremio* se conecta com o *Project Nessie* e com o banco de dados *PostgreSQL*. Com o primeiro, é possível gerenciar os arquivos armazenados no formato *Iceberg* e criar visões de bancos de dados que podem ser consumidas pelos usuários do sistema. Com o segundo, é possível consultar as métricas do sistema para fins de seu monitoramento com o uso de *dashboards*.

As tecnologias citadas são de código aberto e possuem imagens *Docker* públicas (ver [165] e [166]). A implementação do motor de consulta atende aos requisitos RF-5, RF-6, RF-7, RNF-3, RNF-4, e favorece o sistema em seus atributos confiabilidade (AQ-5 e AQ-6) e desempenho (AQ-7).

4.2.10 Ciência de Dados

O componente escolhido é o **JupyterHub**, que se caracteriza por ser um ambiente colaborativo que utiliza *notebooks jupyter* com acesso aos dados armazenados, permitindo que grupos de usuários acessem seus recursos sem a necessidade da instalação de programas em seus computadores [167]. A finalidade é permitir a exploração dos dados e o desenvolvimento de atividades como, por exemplo, análise estatística, mineração de dados e aprendizado profundo. Também é uma tecnologia aberta com imagem *Docker* disponível [168]. Desta forma, o componente está de acordo com os requisitos RF-7, RF-9, RNF-3, RNF-4, assim como contribui para o desempenho (AQ-7) e usabilidade (AQ-8) da arquitetura.

4.2.11 Análise de Dados

O **Apache Superset** foi o componente escolhido para permitir aos usuários explorar e visualizar dados sem a necessidade de escrever código [169]. O componente foi empregado para a elaboração de relatórios *ad-hoc* e exibição de *dashboards* interativos para visualiza-

ção de informações. O *Apache Superset* foi desenvolvido em código aberto e, igualmente, possui imagem *Docker* [170]. Assim, a sua implementação atende aos requisitos RF-7, RF-8, RNF-3, RNF-4, e favorece o sistema em termos de desempenho (AQ-7), e usabilidade (AQ-8).

4.2.12 Geradores de Mensagens

As mensagens nos protocolos DIS e *Protobuf*, usadas para a validação do protótipo são sintetizadas através de geradores de mensagens. Uma arquitetura da suporte a automação de múltiplas sessões de transmissão de mensagens de simulação, no qual um orquestrador controla as instâncias, uma por vez, de acordo com o protocolo testado.

O gerador de mensagens *Protobuf* se conecta diretamente com os coletores na arquitetura, enquanto o **Ouvinte** (ver a Seção 4.2.6) realiza este mesmo papel se o protocolo testado for o DIS. No primeiro caso, a comunicação entre o gerador e o coletor *Protobuf* foi feita por meio de *WebSockets*. No segundo caso, como o DIS utiliza redes locais para a disseminação de mensagens via *broadcast* (ver a Seção 2.2.3), um ouvinte permanece na mesma rede do gerador de mensagens e se comunica também por meio de *WebSockets*, com o coletor dentro da arquitetura.

No orquestrador, arquivos de configuração gerenciam os parâmetros de simulação, como endereço IP de destino, número de iterações, número de mensagens enviadas, intervalo entre cada mensagem, pastas para salvar suas métricas e, mais importante, o protocolo testado. O *script* do orquestrador instancia o *script* de entrada do protocolo a ser testado, que por sua vez se conecta com a arquitetura e inicia a transmissão do número de mensagens assinalado. Ele monitora as saídas padrão (*stdout* e *stderr*) do gerador de mensagens do protocolo alvo, e gerencia as tentativas de reconexão em caso de falha. Cada iteração é concluída quando todas as mensagens foram enviadas com sucesso e um sinal é enviado ao orquestrador, o que permite que o processo seja reiniciado, até que não hajam mais iterações a serem executadas.

Os *scripts* de geração de mensagens também usaram da implementação *Open-DIS* [82] e da biblioteca *Protobufjs* [148] como código de base para desenvolvimento. Para o protocolo DIS foram geradas apenas mensagens do tipo *Entity State PDU*, sendo que somente foram alimentados dados aleatórios referentes aos campos *forceId*, *entityId*, *entityKind*, *domain*, *country*, *category*, *timestamp*, e *entityLocation*. Já para o protocolo *Protobuf* foram geradas mensagens do tipo SimulationClient [171] do software Combater (ver a Seção 2.4.2), preenchidas com dados gerados aleatoriamente para os campos *context*, *clientId*, *knowledge.id*, *knowledgeGroup.id*, *position.latitude*, *position.longitude*, *height_f*, e *pertinence* (*timestamp*).

4.2.13 *Deployment*

O protótipo desenvolvido contou com um ambiente experimental. Assim, foram disponibilizados dois *notebooks* em uma rede local, sendo um para a geração das mensagens e outro para hospedar a arquitetura. O computador com os geradores de mensagens teve como configuração um processador Intel(R) Core(TM) i7-8750H GPU 2.20GHz, 16 GB de RAM, 6 GB de memória de GPU e sistema operacional Windows 11 Pro. O hospedeiro do protótipo apresenta configuração semelhante, com processador Intel(R) Core(TM) i7-8750H CPU 2.21GHz, 16 GB de RAM, 6 GB de memória de GPU e sistema operacional Windows 11 Pro. As placas de vídeo não foram empregadas no experimento. Como um dos requisitos da arquitetura era a utilização de contêiners *Docker*, foi utilizado o *software Docker Desktop* [172] tendo o *Windows Subsystem for Linux (WSL) 2* como *backend* [173].

Além de ser um requisito da arquitetura, o uso de contêineres teve por finalidade permitir a realização de experimentos de forma leve, eficiente e isolada, a fim de emular um ambiente mais controlado. É importante ressaltar que há a recomendação de utilização do orquestrador de *containers Kubernetes* [174] para *deploy* em produção das aplicações *Airflow* [175], *Apache Superset* [176], e *JupyterHub* [177] (ver Seção 4.1.3). A Figura 4.8 apresenta um diagrama com a disposição dos componentes da arquitetura em caso de *deploy* na *EBNet*. O código do protótipo implementado está disponível no *GitHub* [178].

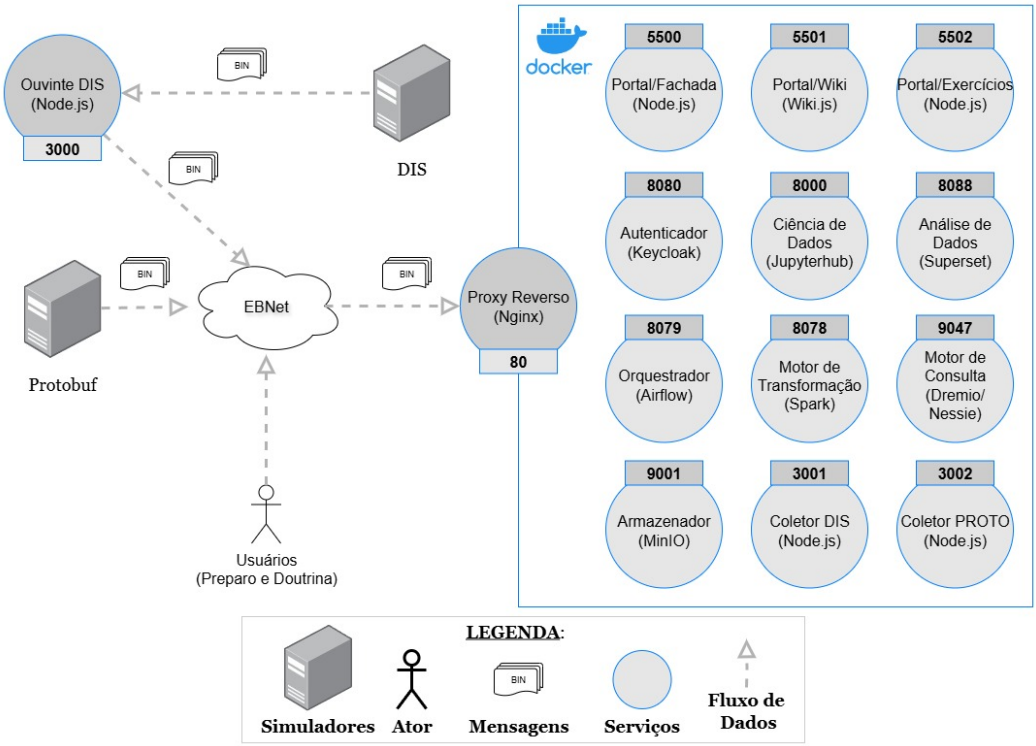

Figura 4.8: Visão de *Deployment*.

4.3 Validação do Protótipo

O processo de validação do protótipo teve por finalidade simular as principais etapas do ciclo de vida da engenharia de dados no contexto apresentado neste trabalho. Primeiramente, foram levantadas as métricas necessárias para a avaliação do protótipo. Em seguida, foram estabelecidos os parâmetros para a execução do experimento. Por último, os resultados foram apresentados e analisados de acordo com as métricas estabelecidas e um parecer a respeito da arquitetura foi emitido.

4.3.1 Metodologia

A ferramenta utilizada para auxiliar no processo de validação da arquitetura proposta foi a abordagem *Goal-Question-Metric (GQM)* [179]. A metodologia propõe o estabelecimento de objetivos norteadores do projeto de desenvolvimento de software, que fomentam a formulação de questionamentos para atestar se tais objetivos são atingidos ou não. Por fim, decidem-se as métricas mais adequadas para responder às questões. Com base nos achados encontrados na análise das métricas estabelecidas, é possível avaliar o protótipo e validar sua adequabilidade quanto ao objetivo final da arquitetura proposta.

O alinhamento do desenho da arquitetura foi estabelecido ao longo do trabalho para auxiliar a aplicação da metodologia escolhida. As arquiteturas de referência e as *frameworks* de simulação tiveram com produto uma série de características desejáveis a uma arquitetura de integração de dados no contexto apresentado (ver Seção 3.4). Estas características orientaram a formulação de Requisitos de Alto Nível (ver a Seção 4.1.2), entre os quais a formulação inicial de um conjunto de Atributos de Qualidade (AQ), que descrevem como o protótipo deve se comportar (ver Tabela 4.4). Ao mesmo tempo, o objetivo final do desenvolvimento da arquitetura é caracterizado pela integração de dados de simulação de combate, e pela extração de valor dos dados coletados pelos sistemas de simulação. O objetivo é apoiar o treinamento militar das tropas e o desenvolvimento da doutrina militar (ver a Seção 4.1.1). Desta forma, o levantamento das métricas encontra-se descrito na Tabela 4.6, e suas definições estão descritas a seguir:

- **Taxa de Sucesso de *Upload***: é o percentual de sucesso durante a movimentação dos dados do *Coletor DIS* ou *Coletor Protobuf* para o *bucket bronze* no *Armazenador*;
- **Taxa de Perdas de Mensagens Coletadas**: é o percentual entre o número de mensagens geradas e o número de mensagens coletadas com sucesso pelo *Coletor DIS* ou pelo *Coletor Protobuf*;

- **Tempo Médio de *Upload***: é o tempo médio em segundos da movimentação dos dados do *Coletor DIS* ou *Coletor Protobuf* para o *bucket Bronze* no *Armazenador*;
- **Taxa de Eficiência do Orquestrador**: é o percentual entre a alocação de recursos do sistema (CPU e memória) no *Orquestrador* (*Apache Airflow*) e a taxa de sucesso das DAGs executadas;
- **Taxa de Eficiência do Motor de Transformação**: é o percentual entre a alocação de recursos do sistema (CPU e memória) no *Motor de Transformação* (*Apache Spark*) e a taxa de sucesso dos *Jobs* executados;
- **Taxa de Eficiência do Motor de Consulta**: é o percentual entre a alocação de recursos do sistema (CPU e memória) no *Motor de Consulta* (*Dremio*) e a taxa de sucesso de consultas realizadas;
- **Taxa de Consistência das Mensagens Coletadas**: é o percentual de igualdade entre o conteúdo das mensagens geradas e o conteúdo mensagens armazenadas no *bucket Warehouse* no *Armazenador*;
- **Taxa de Rastreamento dos Arquivos Armazenados**: é o percentual entre as mensagens geradas e as mensagens transformadas armazenadas no *bucket Warehouse* no *Armazenador*, no formato *Apache Iceberg*;
- **Tempo Médio de Latência do Sistema**: é o tempo médio em milissegundos entre a requisição e a resposta durante a operação do protótipo;
- **Número de Cliques até o Serviço Desejado**: é a soma dos cliques entre a página inicial do sistema e um dos serviços disponibilizados;
- **Taxa de Sucesso de Requisições do Sistema**: é o percentual de requisições bem sucedidas durante a operação do protótipo;
- **Taxa de Eficiência do Sistema**: é o percentual entre a alocação de recursos do sistema (CPU e memória) e a taxa de sucesso global dos principais componentes do sistema (*Orquestrador*, *Motor de Transformação* e *Motor de Consulta*);
- **Número de Incidentes de Segurança**: é a quantidade total de incidentes de violação de segurança durante a operação do sistema;
- **Tempo de Recuperação Após Falhas**: é o tempo em segundos entre uma falha involuntária e o reestabelecimento da operação normal do sistema;
- **Índice de Estabilidade das Respostas**: é o desvio padrão do tempo de resposta durante a operação do sistema;

- **Número de Componentes Ativos**: é a quantidade total de *containers* existente no protótipo;
- **Taxa de Compressão dos Dados Armazenados**: é a proporção entre o espaço total ocupado pelos dados coletados no *bucket Bronze*, e o espaço total ocupado pelos dados armazenados no *bucker Warehouse* no *Armazenador*.

4.3.2 Experimento

A validação da arquitetura foi embasada por medições do sistema durante seu funcionamento. Para isso, foi desenhado um experimento controlado de operação, com enfoque nos níveis mais baixos do ciclo de vida de engenharia de dados (ver Seção 3.2). Neste contexto, a coleta de métricas foi direcionada para a geração, ingestão, armazenamento e transformação de dados, atendendo a demanda motivadora para a realização do trabalho (ver a Seção 1.1), o que representa um enfoque sobre os componentes das camadas de **Ingestão**, **Armazenamento** e **Transformação**. Esta concepção foi orientada pela premissa de que, para que uma organização desenvolva atividade em áreas como *Analytics* e IA, há a necessidade da construção de uma sólida infraestrutura de engenharia de dados, como afirmado por Rogati [1] (ver Seção 1.1).

O experimento simulando a operação do sistema foi composto por uma fase de geração de dados, e outra de transformação e consulta dos dados. Na primeira fase, o protocolo DIS foi usado na geração controlada de mensagens, sendo seguido pelo *Protobuf*, em etapas distintas com 30 iterações cada. Esta fase tinha como foco principal avaliar o funcionamento dos componentes **Coletor DIS** e **Coletor *Protobuf***. Na segunda fase, foram realizadas 30 transformações e 60 consultas para cada grupo de mensagens de simulação salvo. As transformações visavam mensurar as operação do componente **Motor de Transformação** (*Spark*), e as consultas permitiram a extração de métricas do componente **Motor de Consultas** (*Dremio*), e assim verificar o estado do funcionamento da arquitetura a cada dois minutos. O componente **Armazenador** foi observado em ambas as fases, já que permeia as demais camadas.

A geração de dados teve como parâmetro principal a velocidade de uma mensagem gerada a cada milissegundo, conforme descrito por Song *et al.* [79] (ver Seção 3.1). Como mencionado no parágrafo anterior, cada protocolo de comunicação teve 30 iterações de geração de mensagens, cada um com 100.000 mensagens para o protocolo DIS e 20.000 para o *Protobuf*. Cada iteração foi projetada para durar pelo menos 120 segundos com intervalo de 30 segundos, tudo para atender a recomendação da norma RFC 2544 (*Benchmark Methodology for Network Interconnect Devices* [180], que orienta experimentos de

rede devem durar, pelo menos, 60 segundos. Os geradores de mensagens (ver a Seção 4.2.12) tinham por tarefa coletar as seguintes métricas:

- Taxa Média de Geração em Mensagens por Segundo;
- Tamanho Médio de Cada Mensagem em KB;
- Espaço Total dos Dados Gerados em MB;
- Tempo Médio de Cada Iteração em Segundos;
- Total de Todas as Iterações em Minutos;
- Total de Mensagens Geradas.

A ingestão de dados foi realizada de forma automática pelos coletores de mensagens. Ao término de cada iteração de geração, um arquivo *pdf* com uma Ordem de Operações foi associado a um exercício fictício por meio da utilização componente **Portal**. O processo detalhado de funcionamento da coleta e associação está melhor descrito nas Subseções 4.2.4 e 4.2.6.

4.3.3 Resultados

Os resultados obtidos durante a execução do experimento fornecem uma visão quantitativa sobre o desempenho do protótipo implementado, permitindo uma avaliação abrangente da arquitetura proposta. Inicialmente, são apresentados os resultados referentes à geração de mensagens, relacionando-os de forma explícita com as características de *Big Data* (Volume, Velocidade, Variedade, Veracidade, e Valor), conforme discutido na Seção 3.1.

Essa análise destaca como os dados de simulação de combate, simulados no experimento, replicam desafios reais enfrentados em ambientes militares, o que justifica a necessidade de uma arquitetura robusta e escalável. Em seguida, são analisadas as métricas do protótipo, confrontando-as diretamente com as questões estabelecidas na metodologia *Goal-Question-Metric (GQM)*, para avaliar o atendimento aos atributos de qualidade (AQ) da arquitetura. Essa confrontação é feita de maneira sistemática, destacando não apenas os valores observados, mas também suas implicações práticas para o contexto de integração de dados de simulação.

Resultados da Geração de Mensagens

A Tabela 4.7 apresenta as métricas coletadas durante a fase de geração de mensagens para os protocolos DIS e *Protocol Buffers*. Esses resultados destacam o desafio inerente de lidar com dados de simulação de combate, que exibem características típicas de *Big*

Data, exigindo componentes capazes de processar fluxos intensos de informações em tempo real ou próximo ao real. Para contextualizar, o experimento simulou cenários nos quais múltiplas entidades (como veículos ou tropas) geram atualizações constantes, semelhantes a exercícios militares reais, onde dezenas ou centenas de simuladores podem operar simultaneamente.

Para o protocolo DIS, foram geradas 3 milhões de mensagens do tipo *Entity State PDU*, com uma taxa média de geração de 552,29 mensagens por segundo. Cada mensagem possuía um tamanho médio de 160 KB, resultando em um espaço total ocupado de 480 MB ao final das 30 iterações. O tempo médio por iteração foi de 179,90 segundos, com um tempo total de 89,95 minutos para todas as iterações, refletindo a intensidade do processo de geração contínua.

Para o protocolo *Protocol Buffers*, foram geradas 600 mil mensagens do tipo *Simulation Client*, com uma taxa média de 84,79 mensagens por segundo. O tamanho médio por mensagem foi de 55,98 KB, ocupando um espaço total de 33,59 MB. O tempo médio por iteração foi de 238,76 segundos, totalizando 119,38 minutos para as 30 iterações, o que indica uma geração mais demorada devido à complexidade das mensagens *Protobuf*, que são compactas mas exigem mais processamento para serialização.

Esses resultados ilustram as características de *Big Data* da geração de mensagens de simulação, demonstrando como a arquitetura deve ser projetada para lidar com esses aspectos:

- **Volume**: o volume de dados é elevado, considerando os parâmetros do experimento. Foram simuladas 3 milhões de mensagens para DIS e 600 mil para *Protocol Buffers*, e 30 arquivos de contexto tático, perfazendo um total de 509,1 MB em um período de 209,33 minutos (ou aproximadamente 3 horas e meia). Observa-se que o volume pode crescer de forma exponencial num caso extremo como o do experimento, indicando a necessidade de uma arquitetura com armazenamento e transformação escaláveis;
- **Velocidade**: a velocidade de geração é significativa, com taxas de centenas de mensagens por segundo no DIS e dezenas no *Protocol Buffers*. Infere-se a que demanda é elevada, demandando processamento e ingestão rápidos, para evitar perdas ou gargalos na arquitetura;
- **Variedade**: a diversidade é observada na utilização de arquivos *pdf*, e diferentes protocolos e tipos de mensagens (*Entity State PDU* vs. *Simulation Client*), que incluem dados estruturados como posições geográficas, identificadores e *timestamps*, representando a heterogeneidade típica de fontes de simulação. Também é título de

nota que os exercícios de simulação de combate podem apresentar grande variedade de tipos de dados e de arquivos, como visto na Seção 3.1;

- **Veracidade**: embora os dados sejam gerados sinteticamente para o experimento, a consistência entre geração e coleta (como verificado nas métricas subsequentes) sugere alta veracidade, essencial para análises confiáveis em contextos militares;
- **Valor**: reside na capacidade de extrair *insights* para treinamento de tropas e desenvolvimento doutrinário, como trajetórias de entidades ou análises de contextos doutrinários complexos. Não foi observado diretamente pelo experimento porque exige maior relação entre os dados de simuladores e o contexto operacional. Entretanto, é possível inferir que a transformação de dados brutos pode produzir conhecimento doutrinário relevante.

Os achados quanto à geração de dados confirma que há exposição do protótipo aos aspectos do *Big Data* no contexto militar. As decisões arquiteturais e de implementação visam superar os rigores do cenário de experimentação proposto, garantindo que o protótipo seja capaz de suportar as demandas reais de apoiar exercícios táticos que utilizem a simulação de combate.

Análise das métricas do protótipo

A Tabela 4.8 resume as métricas coletadas durante a operação do protótipo, abrangendo aspectos como sucesso de operações, eficiência de recursos e latência. A seguir, essas métricas são confrontadas com as questões definidas na metodologia *GQM* (ver Tabela 4.6), avaliando o atendimento a cada atributo de qualidade. Cada questão é analisada com base nas métricas associadas, destacando o desempenho observado, possíveis limitações e implicações para a arquitetura em um contexto de treinamento e doutrina militar. Assim, os resultados da análise das métricas do protótipo foram as seguintes:

- **O sistema coleta e armazena eventos locais com eficácia?** (AQ-1, Desempenho): com Taxa de Sucesso de *Upload* de 100% e Taxa de Perda de Mensagens Coletadas de apenas 1,0004%, o sistema demonstra eficácia na coleta e armazenamento inicial. Isso indica que o *Coletor DIS* e o *Coletor Protobuf*, além do *Armazenador* (*MinIO*) operam de forma robusta, alinhando-se ao requisito de lidar com alta velocidade de dados sem perdas significativas;
- **O sistema move os dados armazenados localmente com baixa latência?** (AQ-2, Desempenho): o Tempo Médio de *Upload* de 0,533 ms é extremamente baixo, confirmando baixa latência na movimentação de dados para o *bucket Bronze*. Essa métrica valida a eficiência da camada de ingestão, especialmente em cenários de alta

velocidade, evitando atrasos que poderiam comprometer a integridade de simulações em tempo real;

- **O sistema move os dados armazenados com elevada eficiência?** (AQ-3, Desempenho): a Taxa de Eficiência do *Orquestrador* de 277,57% (superior a 100%, indicando subutilização de recursos com alto sucesso nas DAGs) demonstra elevada eficiência na coordenação de movimentações via *Airflow*. Isso sugere que o componente gerencia fluxos com sobra de capacidade, favorecendo escalabilidade em ambientes com múltiplos simuladores conectados simultaneamente;
- **O sistema transforma dados armazenados com elevada eficiência?** (AQ-4, Desempenho): com Taxa de Eficiência do *Motor de Transformação* de 708,47%, o *Spark* processa transformações com alta eficiência, lidando com grandes volumes (como os 480 MB de DIS) de forma otimizada por meio de processamento paralelo. Isso confirma o alinhamento com as características de *Big Data*, permitindo transformações rápidas que preparam dados para análises avançadas;
- **O sistema realiza consultas com elevada eficiência?** (AQ-5, Desempenho): a Taxa de Eficiência do *Motor de Consulta* de 95,25% indica eficiência elevada no *Dremio*, permitindo consultas rápidas em dados transformados. Entretanto, há potencial para otimização em cenários de maior carga;
- **O sistema armazena dados com consistência?** (AQ-6, Confiabilidade): a Taxa de Perda de Mensagens Coletadas de 1,0004% combinada com Taxa de Consistência das Mensagens Coletadas de 100% assegura alta consistência, garantindo que os dados armazenados no *Bucket Warehouse* reflitam fielmente os gerados. Tal aspecto contribui para a veracidade dos dados, levando a análises mais precisas;
- **O sistema armazena dados com rastreabilidade?** (AQ-7, Confiabilidade): a Taxa de Rastreamento dos Arquivos Armazenados de 100% valida o uso do *Iceberg* e *Nessie*, permitindo rastreio completo via metadados e versões, alinhando-se a requisitos de integridade e auditoria, como em Análises Pós-Ação ou análises de grupos de exercícios;
- **A plataforma *web* funciona com reduzida latência?** (AQ-8, Desempenho): o Tempo Médio de Latência do Sistema de 0,1891 ms é mínimo, confirmando operação fluida da plataforma *web*, incluindo o *Portal* e os demais serviços da arquitetura;
- **O sistema monitora o seu estado com elevada eficiência?** (AQ-9, Desempenho): a Taxa de Eficiência do Orquestrador de 277,57% demonstra monitoramento eficiente via *Airflow*, coletando métricas de componentes sem sobrecarga, permitindo detecção proativa de problemas;

- **O acesso aos serviços da plataforma é simples e intuitivo?** (AQ-10, Usabilidade): o Número de Cliques até o Serviço Desejado é de três cliques para usuários logados, e Tempo Médio de Latência de 0,1891 ms. Com isso, o acesso é intuitivo e rápido, favorecendo usabilidade para usuários com variados níveis de expertise técnica;
- **O sistema permanece constantemente disponível e operante?** (AQ-11, Disponibilidade): a Taxa de Sucesso de Requisições do Sistema de 99,54% indica alta disponibilidade, com mínimas falhas observadas, essencial para a operação contínua da arquitetura;
- **O sistema pode receber mais pontos de integração?** (AQ-12, Escalabilidade): a Taxa de Compressão dos Dados Armazenados de 24,48 vezes, e Taxa de Eficiência do Sistema de 360,43% sugerem capacidade para mais integrações, com armazenamento otimizado e eficiência global alta, permitindo expansão para novos sistemas de simulação ou serviços de apoio. É importante ressaltar a restrição de *deploy* para produção de alguns serviços da arquitetura como mencionado na Seção 4.2.13;
- **O sistema opera sem incidentes de segurança?** (AQ-13, Segurança): o Número de Incidentes de Segurança foi de zero durante o experimento. Embora os parâmetros do experimento fossem muito restritivos, a utilização de componentes como o *Autenticador* (*Keycloak*) e o *Proxy Reverso* (*Nginx*) contribuem para uma operação segura do sistema. Para um levantamento mais completo seria necessário um experimento mais longo e sob condições menos restritivas;
- **O sistema se recupera de falhas de operação rapidamente?** (AQ-14, Confiabilidade): o Tempo de Recuperação Após Falhas não foi observado (sem falhas involuntárias), mas a arquitetura conteinerizada sugere recuperação rápida via reinício de *containers*, minimizando *downtime*. Para uma melhor avaliação da métrica seria necessário um período maior de operação e condições de rede menos restritivas;
- **O sistema opera com máxima estabilidade?** (AQ-15, Confiabilidade): com Taxa de Sucesso de Requisições de 99,54% e o Índice de Estabilidade das Respostas de 0,22611 ms (ou baixo desvio padrão). Sob as condições apresentadas, a arquitetura exibiu alta estabilidade, garantindo previsibilidade em uso;
- **O sistema opera com capacidade de substituir componentes?** (AQ-16, Manutenibilidade): o Número de Componentes Ativos de 18 *containers* e Taxa de Compressão dos Dados Armazenados de 24,48 vezes facilitam manutenção, permitindo substituições modulares sem impacto significativo no armazenamento. É

importante ressaltar a necessidade de um mecanismo de orquestração de *containers* como explicado na Seção 4.2.13.

Considerações Finais

Os resultados validam positivamente a arquitetura proposta, demonstrando que o protótipo atende aos requisitos de alto nível e atributos de qualidade estabelecidos, com métricas que superam expectativas em eficiência e confiabilidade. Com desempenho robusto em lidar com características de *Big Data*, o sistema integra dados de simulação de combate de forma eficaz, suportando o objetivo de extrair valor para treinamento militar e desenvolvimento doutrinário. Embora o experimento apresente parâmetros restritos, ainda é possível observar o desempenho da arquitetura face aos requisitos propostos, mesmo em um ambiente controlado. Os achados indicam robustez para cenários reais, sugerindo que a arquitetura é adequada, escalável e pronta para evolução para ambientes de produção como na *EBNet*. É recomendável a realização de testes de maior escala e com condições mais próximas das encontradas em produção para confirmar a resiliência da arquitetura perante sua operação no âmbito da Força Terrestre (F Ter).

Tabela 4.6: Metodologia de estabelecimento de métricas.

Objetivo: Integrar e extrair valor de dados de simulação de combate para apoiar treinamento de tropas e o desenvolvimento da doutrina militar.		
Questão	**Atributo**	**Métricas**
O sistema coleta e armazena eventos locais com eficácia?	Desempenho (AQ-1)	Taxa de Sucesso de *Upload*, Taxa de Perda de Mensagens Coletadas
O sistema move os dados armazenados localmente com baixa latência?	Desempenho (AQ-2)	Tempo Médio de *Upload*
O sistema move os dados armazenados com elevada eficiência?	Desempenho (AQ-3)	Taxa de Eficiência do Orquestrador
O sistema transforma dados armazenados com elevada eficiência?	Desempenho (AQ-4)	Taxa de Eficiência do Motor de Transformação
O sistema realiza consultas com elevada eficiência?	Desempenho (AQ-5)	Taxa de Eficiência do Motor de Consultas
O sistema armazena dados com consistência?	Confiabilidade (AQ-6)	Taxa de Perda de Mensagens Coletadas, Taxa de Consistência das Mensagens Coletadas
O sistema armazena dados com rastreabilidade?	Confiabilidade (AQ-7)	Taxa de Rastreamento de Arquivos Armazenados
A plataforma *web* funciona com reduzida latência?	Desempenho (AQ-8)	Tempo de Médio de Latência do Sistema
O sistema monitora o seu estado com elevada eficiência?	Desempenho (AQ-9)	Taxa de Eficiência do Orquestrador
O acesso aos serviços da plataforma é simples e intuitivo?	Usabilidade (AQ-10)	Número de Cliques até o Serviço Desejado, Tempo de Médio de Latência do Sistema
O sistema permanece constantemente disponível e operante?	Disponibilidade (AQ-11)	Taxa de Sucesso de Requisições do Sistema
O sistema pode receber mais pontos de integração?	Escalabilidade (AQ-12)	Taxa de Compressão de Dados Armazenados, Taxa de Eficiência do Sistema
O sistema opera sem incidentes de segurança?	Segurança (AQ-13)	Número de Incidentes de Segurança
O sistema se recupera de falhas de operação rapidamente?	Confiabilidade (AQ-14)	Tempo de Recuperação Após Falhas
O sistema opera com máxima estabilidade?	Confiabilidade (AQ-15)	Taxa de Sucesso de Requisições do Sistema, Índice de Estabilidade das Respostas
O sistema opera com capacidade de substituir componentes?	Manutenibilidade (AQ-16)	Número de Componentes Ativos, Taxa de Compressão dos Dados Armazenados

Tabela 4.7: Métricas de geração de mensagens.

Métrica	**DIS**	***Protocol Buffers***
Mensagem	*Entity State PDU*	*Simulation Client*
Taxa média de geração de mensagens (Msg/seg)	552,29	84,79
Tamanho médio de cada mensagem (KB)	160	55,98
Espaço total dos dados gerados (MB)	480	33,59
Tempo médio de cada iteração (seg)	179,90	238,76
Tempo total de todas as iterações (min)	89,95	119,38
Total de mensagens geradas	3 milhões	600 mil

Tabela 4.8: Métricas do Protótipo.

Métrica	**Valor**
Taxa de Sucesso de *Upload*	100%
Taxa de Perda de Mensagens Coletadas	1,0004%
Tempo Médio de *Upload*	0,533 ms
Taxa de Eficiência do Orquestrador	277,57%
Taxa de Eficiência do Motor de Transformação	708,47%
Taxa de Eficiência do Motor de Consulta	95,25%
Taxa de Consistência das Mensagens Coletadas	100%
Taxa de Rastreamento dos Arquivos Armazenados	100%
Tempo Médio de Latência do Sistema	0,1891 ms
Número de Cliques até o Serviço Desejado	3 cliques
Taxa de Sucesso de Requisições do Sistema	99,54%
Taxa de Eficiência do Sistema	360,43%
Número de Incidentes de Segurança	0 incidentes
Tempo de Recuperação Após a Falha	Não observado
Índice de Estabilidade das Respostas	0,22611 ms
Número de Componentes Ativos	18 *containers*
Taxa de Compressão dos Dados Armazenados	24,48 vezes

Capítulo 5

Conclusão

Este trabalho teve como objetivo propor e validar uma arquitetura distribuída para integrar dados gerados por sistemas de simulação de combate no âmbito da Força Terrestre (F Ter) do Exército Brasileiro (EB), visando suportar a aplicação de Ciência de Dados e Inteligência Artificial (IA) no treinamento e na experimentação doutrinária. A pesquisa partiu da necessidade de uma infraestrutura robusta de engenharia de dados para viabilizar a extração de valor a partir dos dados de simulação, superando a ausência de mecanismos unificados para coleta, armazenamento e transformação.

Os objetivos específicos foram alcançados com êxito. Inicialmente, foi proposta uma arquitetura em camadas, baseada em princípios de engenharia de dados e modelos de referência, capaz de integrar os diferentes tipos de simulação (viva, virtual e construtiva). A validação foi realizada por meio da metodologia *Goal-Question-Metric (GQM)*, aferindo métricas de qualidade, como eficiência, confiabilidade e escalabilidade. Além disso, a arquitetura demonstrou aderência às características de *Big Data* (volume, velocidade, variedade, veracidade e valor), confirmando sua aplicabilidade ao contexto da simulação de combate.

Os testes experimentais, conduzidos em ambiente controlado, evidenciaram a robustez da solução, que lidou com altos volumes de dados, taxas elevadas de geração de mensagens e diversidade de formatos, mantendo baixa latência e alta confiabilidade nos processos de ingestão, transformação e consulta. Embora realizados com parâmetros restritivos, os resultados sugerem que a arquitetura é escalável e pode ser aplicada em cenários reais, como na *EBNet*. Recomenda-se, no entanto, a realização de testes em maior escala, em condições operacionais, para consolidar a resiliência da solução e identificar possíveis ajustes.

Como contribuição, este trabalho apresenta uma proposta pioneira de integração de dados de simulação de combate, fundamentada em conceitos de Modelagem e Simulação, e *Big Data*, com viabilidade técnica e valor estratégico para EB. Como perspectivas futuras,

destacam-se a aplicação em *analytics* e IA, além da expansão da arquitetura para outros entes da *Tríplice Hélice*, o que potencializa a simulação como vetor de inovação militar e nacional.

Os trabalhos futuros visualizados para continuidade da presente pesquisa poderão ser realizados em três direções diferentes. Na primeira, poderiam ser realizados *benchmarks* de desempenho visando encontrar uma forma otimizada de armazenar os dados da simulação e de contexto no armazenador de objetos. Na segunda direção, buscar-se-ia utilizar o *Protocol Buffers* como protocolo de comunicação e armazenamento das mensagens de simulação, tudo com a finalidade de fazer a padronização o *pipeline* de dados, facilitando a sua operação e manutenção. Por último, seriam realizados estudos para aperfeiçoamento e expansão da arquitetura de dados visando sua utilização em operações militares correntes com emprego de tropa em todo o território nacional.

Em síntese, a arquitetura proposta atende às necessidades de integração e análise de dados, transformando-os em ativos estratégicos. A consolidação de uma infraestrutura de dados robusta permitirá ao EB avançar na utilização de tecnologias emergentes, fortalecendo o preparo da F Ter e a experimentação doutrinária frente aos desafios do século XXI.

Referências

[1] Rogati, Monica: *The ai hierarchy of needs*, 2017. https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007, acesso em 14/06/2024. xi, 1, 2, 74

[2] Defesa Aérea & Naval. https://www.defesaaereanaval.com.br/exercito/aviacao-do-exercito-utiliza-o-simulador-de-helicopteros-esquilo-e-fennec, acesso em 2024/05/01. xi, 6

[3] 2ª Divisão de Exército. https://2de.eb.mil.br/index.php/ultimas-noticias/2180-2-divisao-de-exercito-participa-do-exercicio-de-simulacao-construtiva. xi, 7

[4] Revista Asas. Edição Online. https://www.edrotacultural.com.br/exercito-brasileiro-se-prepara-para-exercicio-conjunto-com-os-eua/, acesso em 2024/05/01. xi, 7

[5] Tolk, Andreas: *Engineering Principles of Combat Modeling and Distributed Simulation*, páginas 61–63, 190, 347, 351–352, 348–349. Wiley & Sons, Estados Unidos, 2012. xi, 8, 9, 10, 11, 12, 13

[6] Defesa Em Foco. https://www.defesaemfoco.com.br/brigada-de-infantaria-motorizada-faz-simulacao-virtual-de-combate/, acesso em 2024/05/01. xi, 19

[7] Real And Simulated Wars. https://kriegsimulation.blogspot.com/2021/01/steel-beasts-prope-in-game-aar-feature.html, acesso em 2024/05/01. xi, 20

[8] 3ª Divisão de Exército. https://3de.eb.mil.br/index.php/todas-as-noticias/1532-epex-visita-a-ufsm, acesso em 2024/05/01. xi, 20

[9] Forças Terrestres. https://www.forte.jor.br/2017/07/19/exercicio-em-simulador-de-apoio-de-fogo-simaf/, acesso em 2024/05/01. xi, 21

[10] 32º Grupo de Artilharia de Campanha: *Grupo d. pedro i realiza pci no simulador bombarda*, 2021. http://www.32gac.eb.mil.br/index.php/noticias/244-grupo-d-pedro-i-inaugura-sala-do-simulador-bombarda, acesso em 06/08/2021. xi, 21

[11] Defesa Aérea & Naval. https://www.defesaaereanaval.com.br/aviacao/aviacao-do-exercito-inaugura-sala-de-simuladores-de-voo-modernizados, acesso em 2024/05/01. xi, 22

[12] Reis, Joe e Matt Housley: *Fundamentals of Data Engineering: Plan and Build Robust Data Systems*, páginas 2, 12, 40–42, 43–44, 48–65, 156–164, 197–213, 197–213, 215–217. O'Reilly, 2022. xi, 2, 28, 29, 30, 32, 33, 34, 36

[13] Sokolowski, John A. e Catherine M. Banks: *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains*, páginas 12–14, 20–21, 373–374, 388. Wiley & Sons, Estados Unidos, 2010. xii, 5, 8, 9, 10, 12, 28

[14] Layton, Peter: *Algorithmic Warfare: Applying Artificial Intelligence to Warfighting*, página 1. 2018. 1

[15] Reding, D.F. e J. Eaton: *Science & Technology Trends 2020-2040*, páginas 6–7. 2020. https://www.nato.int/nato_static_fl2014/assets/pdf/2020/4/pdf/190422-ST_Tech_Trends_Report_2020-2040.pdf. 1

[16] Chefe do Estado-Maior do Exército: *Portaria - C Ex nº 1.318, de 14 de abril de 2024. Aprova a Diretriz Estratégica de Inteligência Artificial para o Exército Brasileiro.* 2024. http://www.sgex.eb.mil.br/sg8/006_outras_publicacoes/01_diretrizes/04_estado-maior_do_exercito/port_n_1318_eme_14mai2024.html. 1

[17] H.W., Meerveld, Lindelauf R.H.A. e Postma E.O: *The irresponsibility of not using AI in the military*, página 1. 2023. https://doi.org/10.1007/s10676-023-09683-0. 1

[18] Maslow e Abraham H.: *A theory of human motivation.* Psychological Review, 50(4):370–396, 1943. http://doi.org/10.1037/h0054346. 2

[19] Defense, Secretary of: *DoD Modeling and Simulation (M&S) Glossary*, página 136 e 157. 1998. 4

[20] Birta, Loius G. e Gilbert Arbez: *Modeling and Simulation: Exploring Dynamic System Behaviour*, página vii. Springer Nature, Suíça, 2019. 5

[21] Ören, Tuncer, Bernard P. Zeigler e Andreas Tolk: *Body of Knowledge for Modeling and Simulation: A Handbook by the Society for Modeling and Simulation International*, páginas 2, 122, 150 e 160–162. Springer Nature, Suíça, 2023. 5, 6, 7, 10, 12

[22] Exército Brasileiro: *Caderno de Instrução Emprego da Simulação. Edição Experimental. Item 3.2.3*. 2020. http://www.sgex.eb.mil.br/sg8/004_documentos_instrucao/01_cadernos_de_instrucao(ci)/port_n_133_coter_02out2020.html. 5, 6, 7

[23] Exército Brasileiro: *Caderno de Instrução Exercícios de Simulação Virtual. Item 1.2.9*. 2020. http://www.sgex.eb.mil.br/sg8/004_documentos_instrucao/01_cadernos_de_instrucao(ci)/port_n_134_coter_02out2020.html. 6

[24] Exército Brasileiro: *Caderno de Instrução Exercícios de Simulação Construtiva. Itens 1.3.1, 1.3.2, 1.3.9 e 1.3.10*. 2017. http://www.sgex.eb.mil.br/sg8/004_documentos_instrucao/01_cadernos_de_instrucao(ci)/port_n_018_coter_08maio2017.html. 6

[25] Çayirci, Erdal e Dusan Marincic: *Computer Assisted Exercises and Training: A Reference Guide*, página 183. Wiley & Sons, Estados Unidos, 2009. 6

[26] Exército Brasileiro: *Caderno de Instrução Exercícios com Emprego da Simulação Viva. Edição Experimental. Itens 1.3.4 e 1.3.17*. 2021. http://www.sgex.eb.mil.br/sg8/004_documentos_instrucao/01_cadernos_de_instrucao(ci)/port_n_100_coter_23ago2021.html. 7, 23

[27] Open-DIS: *About open-dis. an open source implementation of the ieee-1278.1 distributed interactive simulation (dis) application protocol.*, 2024. https://open-dis.org/, acesso em 16/03/2024. 13, 61, 68

[28] IEEE Standards Association: *IEEE Std 1278.1-2012. Standard for Distributed Interactive Simulation - Application Protocols*, página xiii. Estados Unidos, 2012. 13

[29] *Unity technologies*. https://unity.com/pt, acesso em 2024/03/01. 13

[30] *Unreal engine*. https://www.unrealengine.com/pt-BR, acesso em 2024/03/01. 13

[31] Gaming Research Integration for Learning Laboratory (GRILL): *Opendis plugin*, 2024. https://www.gamingresearchintegrationforlearninglab.com/projects/opendis, acesso em 16/03/2024. 13

[32] Congresso Nacional: *Constituição da República Federativa do Brasil. Artigo 142 (Caput)*. 1988. https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm. 14

[33] Congresso Nacional: *Lei Complementar nº 97, de 9 de junho de 1999. Dispõe sobre as normas gerais para a organização, o preparo e o emprego das Forças Armadas, Artigo 13*. 1999. https://www.planalto.gov.br/ccivil_03/leis/lcp/lcp97.htm. 14

[34] Congresso Nacional: *Lei Complementar nº 117, de 2 de setembro de 2004. Altera a Lei Complementar nº 97, de 9 de junho de 1999, que dispõe sobre as normas gerais para a organização, o preparo e o emprego das Forças Armadas, para estabelecer novas atribuições subsidiárias*. 2004. https://www.planalto.gov.br/ccivil_03/leis/lcp/lcp117.htm. 14

[35] Congresso Nacional: *Lei Complementar nº 136, de 25 de agosto de 2010. Altera a Lei Complementar nº 97, de 9 de junho de 1999, que dispõe sobre as normas gerais para a organização, o preparo e o emprego das Forças Armadas, para criar o Estado-Maior Conjunto das Forças Armadas e disciplinar as atribuições do Ministro de Estado da Defesa*. 2010. https://www.planalto.gov.br/ccivil_03/leis/lcp/Lcp136.htm. 14

[36] Exército Brasileiro: *Portaria - C Ex nº 2.147, de 20 de dezembro de 2023. Aprova a Política Militar Terrestre - Fase 3 do Sistema de Planejamento Estratégico do Exército para o Ciclo 2024-2027 (EB10-P-01.016), 1ª edição. Item 3.2.2.1.* 2023. http://www.sgex.eb.mil.br/sg8/006_outras_publicacoes/05_politicas/port_n_2147_cmdo_eb_20dez2023.html. 14

[37] Exército Brasileiro: *Portaria - C Ex nº 2.148, de 20 de dezembro de 2023. Aprova a Concepção Estratégica do Exército (Plano) - integrante da Fase 4 do Sistema de Planejamento Estratégico do Exército para o Ciclo 2024-2027 (EB10-P-01.017), 1ª edição. Items 2.5.10.2, 5.1.1 e 5.1.3.* 2023. http://www.sgex.eb.mil.br/sg8/006_outras_publicacoes/04_planos/port_n_2148_cmdo_eb_20dez2023.html. 15

[38] Comando de Operações Terrestres: *Portaria - COTER nº 219, de 13 de novembro de 2019. Aprova a Diretriz Organizadora do Sistema de Prontidão Operacional da Força Terrestre (SISPRON) e dá outra providência. Letras "f" do Item 5, e "a" e "b" do Item 6.* 2019. http://www.sgex.eb.mil.br/sg8/006_outras_publicacoes/01_diretrizes/02_comando_de_operacoes_terrestres/port_n_219_coter_13nov2019.html. 15

[39] Exército Brasileiro: *Portaria - COTER nº 348 (Acesso Restrito), de 18 de outubro de 2023. Aprova o Programa de Instrução Militar (EB70-P-11.001), para o ano de 2024, e dá outras providências. Items 5.2.1, 5.2.2, 5.3.4, 5.3.5, 5.3.6, 5.3.7, e 5.4.3.* 2024. 15, 16

[40] Centro de Adestramento - Sul: *Missão e visão de futuro*, 2024. https://casul.eb.mil.br/index.php/missao-e-visao-de-futuro, acesso em 16/03/2024. 16

[41] Centro de Adestramento - Leste: *Visão de futuro*, 2024. https://www.caleste.eb.mil.br/visao-de-futuro.html, acesso em 16/03/2024. 16

[42] Centro de Instrução de Blindados: *Missão*, 2024. https://cibld.eb.mil.br/index.php/missao, acesso em 16/03/2024. 17

[43] Centro de Instrução de Aviação do Exército: *Missão, visão de futuro e valores*, 2024. https://ciavex.eb.mil.br/index.php/missao-visao-de-futuro-e-valores, acesso em 16/03/2024. 17

[44] Centro de Instrução de Artilharia de Mísseis e Foguetes: *Missão*, 2024. http://www.ciartmslfgt.eb.mil.br/index.php/5, acesso em 16/03/2024. 17

[45] Academia Militar das Agulhas Negras: *Missão, visão e valores*, 2023. https://www.aman.eb.mil.br/missao-visao-e-valores, acesso em 08/05/2023. 17

[46] Comando de Operações Terrestres: *Portaria - COTER nº 002, de 12 de abril de 2018. Aprova as Instruções Reguladoras da Sistemática de Experimentação Doutrinária (EB70-IR-10.002), 1ª Edição. Arts. 8º, 9º, 29º e 30º".* 2018. https://bdex.eb.mil.br/jspui/handle/1/1170. 17

[47] Department of Defense: *Experimentation Guidebook: Prototypes and Experiments*, página 3. 2021. 17

[48] The Technical Cooperation Program: *Guide for Understanding and Implementing Defense Experimentation (GUIDEx)*, páginas vii–ix. 2006. 17

[49] Department of The Army: *FM 3-0: Operations*, páginas 1–16,1–17. 2022. https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN36290-FM_3-0-000-WEB-2.pdf. 18

[50] Department of The Army: *Army Modeling and Simulation (M&S): Final Report*, páginas 3–5. 2021. https://asb.army.mil/Reports/. 18

[51] Neto, Jerson Geraldo: *Emprego do software virtual battlespace simulator 3 como ferramenta de apoio ao ensino dos cadetes da aman*. Revista Agulhas Negras, página 159, 2022. http://www.ebrevistas.eb.mil.br/aman/article/view/10044. 19

[52] Centro de Adestramento - Sul: *Missão e visão de futuro*, 2023. https://casul.eb.mil.br/index.php/ultimas-noticias/550-estagio-de-administrador-do-simulador-virtual-tatico-vbs3, acesso em 04/10/2023. 19

[53] Bohemia Interactive Simulations: *VBS Gateway*, página 6. 2019. 19

[54] Soares, Edilmar Schmacker: *O emprego dos simuladores virtuais táticos no adestramento de forças tarefas blindadas*. Revista Ação de Choque, páginas 35–36, 2015. http://www.ebrevistas.eb.mil.br/AC/article/view/2866. 19

[55] Calytrix Technologies: *Systems Implementation Primer for DIS/HLA Simulations*, páginas 1–21. 2014. 19

[56] Siqueira, Iago Capanema: *Proposta de modelo de avaliação no simulador virtual tático com base nos simuladores utilizados no exército brasileiro*. página 13 e 28, 2022. http://bdex.eb.mil.br/jspui/handle/123456789/11461. 20

[57] Domenico, Gabriel Di: *Desenvolvimento de Mini-Mapa Interativo e Interfaces Naturais de Usuário para Navegação em Aplicações de Realidade Virtual com Terrenos de Grande Escala*, página 10. 2023. https://repositorio.ufsm.br/handle/1/31111. 20

[58] Wilin, Leonel Francisco Slim: *O simulador de apoio de fogo e a influência no adestramento de tropas*. Revista Doutrina Militar Terrestre, página 57, 2021. http://www.ebrevistas.eb.mil.br/DMT/article/view/8931. 20

[59] Revista Ecobravo: *Bombarda: Uso inédito do simulador virtual no adestramento do artilheiro*, 2022. http://www.ebrevistas.eb.mil.br/ECO/article/view/6395/5534, acesso em 22/10/2022. 21

[60] 5º Grupo de Artilharia de Campanha Autopropulsado: *Visita de orientação técnica da ad/5*, 2024. https://5gacap.eb.mil.br/index.php/atividades-da-om-2022/417-visita-de-orientacao-tecnica-da-ad-5, acesso em 16/03/2024. 21

[61] Comando Militar no Nordeste: *Estágio de condução do tiro de artilharia no 7º grupo de artilharia de campanha*, 2020. https://cmne.eb.mil.br/ultimas-noticias/estagio-de-conducao-do-tiro-de-artilharia-no-7-grupo-de-artilharia-de-campanha, acesso em 15/10/2020. 21

[62] 28º Grupo de Artilharia de Campanha: *28º gac inaugura a sala de simulação de tiro de artilharia*, 2022. https://28gac.eb.mil.br/index.php/ultimas-noticias/1245-28-gac-inaugura-a-sala-de-simulacao-de-tiro-de-artilharia, acesso em 27/09/2022. 21

[63] 3ª Divisão de Exército: *19º gac é inspecionado pelo comandante militar do sul*, 2024. https://3de.eb.mil.br/index.php/todas-as-noticias/2427-19-gac-e-inspecionado-pelo-comandante-militar-do-sul, acesso em 16/03/2024. 21

[64] Associação Brasileira das Indústrias de Materiais de Defesa e Segurança: *Operação sisson e as novas tecnologias em artilharia*, 2021. https://abimde.org.br/en/noticias/operacao-sisson-e-as-novas-tecnologias-em-artilharia/, acesso em 01/06/2021. 21

[65] Defesanet: *Ca-sul - projeto bombarda*, 2019. https://www.defesanet.com.br/terrestre/ca-sul-projeto-bombarda/, acesso em 19/05/2019. 21

[66] Agência Nacional de Aviação Civil (ANAC): *Definições e normas*, 2021. https://www.gov.br/anac/pt-br/assuntos/regulados/empresas-aereas/simuladores-de-voo-fstd/definicoes-e-normas, acesso em 01/12/2021. 22

[67] Rocha, Leonard Soares da: *O Emprego de Dispositivos de Simulação de Voo no Adestramento Tático dos Pelotões de Reconhecimento e Ataque da Aviação do Exército para as Missões de Combate*, páginas 72–82. 2017. http://bdex.eb.mil.br/jspui/handle/1/1105. 22

[68] Air Force Research Lab. Warfighter Readiness Division: *Evaluation of Game-Based Visualization Tools for Military Flight Simulation*, páginas 7–9. 2017. 22

[69] Silva Júnior, Ersino Albano da: *O centro de adestramento sul: Uma nova ferramenta para o preparo da tropa.* Revista Doutrina Militar Terrestre, página 50, 2019. http://www.ebrevistas.eb.mil.br/DMT/article/view/2986. 22

[70] Amorim, Rodolfo Leonardo Borges Carneiro e Anderson Wallace de Paiva dos Santos: *As inovações tecnológicas de simulação aplicada no processo ensino-aprendizagem: A experiência do exército brasileiro.* Revista Military Review, página 5, 2022. https://www.armyupress.army.mil/Journals/Edicao-Brasileira/Artigos-Exclusivamente-On-line/Artigos-Exclusivamente-On-line-de-2022/Amorim-POR-OLE-Jan-2022/. 22

[71] MASA: *MASA SWORD: Módulo HLA*, página 4. 2017. 22

[72] Mello, Fellipe Brum: *Implementação de Um Cliente Remoto para o Simulador MASA Sword*, páginas 22–34. 2016. https://repositorio.ufsm.br/handle/1/24841. 22, 30

[73] SAAB: *A simulação viva nos exercícios de preparação do ca-leste*, 2024. https://www.saab.com/pt-br/markets/brasil/historias/2023/a-simulacao-viva-nos-exercicios-de-preparacao-do-ca-leste, acesso em 16/03/2024. 23

[74] SAAB: *Simulação viva: Realismo para resultados efetivos*, 2024. https://www.saab.com/pt-br/markets/brasil/historias/2020/simulacao-viva-realismo-para-resultados-efetivos, acesso em 16/03/2024. 23

[75] NATO Science and Technology Organization: *Urban Combat Advanced Training Technology Architecture*, páginas 1–9. 2018. https://apps.dtic.mil/sti/pdfs/AD1047399.pdf. 23

[76] Underdahl, Brian: *Data Integration for Dummies*, páginas 16–17. Wiley & Sons, 2018. 25

[77] Elmasri, Ramez e Shamkant B. Navathe: *Fundamentals of Database Systems*, página 914. Pearson, 2016. 26

[78] Jukic, Nenad, Susan Vrbsky e Abhishek Sharma Svetlozar Nestorov: *Database Systems: Introduction to Databases and Data Warehouses*, página 367. Prospect Press, 2021. 26

[79] Song, Xiao, Yulin Wu, Yaofei Ma, Yong Cui e Guandhong Gong: *Military simulation big data: Background, state of the art, and challenges*. Mathematical Problems in Engineering, 2015:3–6, 2015. http://dx.doi.org/10.1155/2015/298356. 26, 27, 74

[80] Liu, XioRui, Juan Ospina, Ioannis Zografopoulos, Alonzo Russel e Charalambos Konstantinou: *Faster than real-time simulation: methods, tools, and applications*. Em *Proceedings of the 9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, New York, NY, USA, 2021. Association for Computing Machinery, ISBN 9781450386081. https://doi.org/10.1145/3470481.3472703. 27

[81] Underdahl, Brian: *Data Integration for Dummies*, página 13. Wiley & Sons, 2018. 28

[82] *node-disnetworkclient*. https://github.com/keckxde/node-disnetworkclient, acesso em 31/05/2024. 30, 61, 70

[83] IEEE Standards Association: *IEEE Std 1278.1-2012. Standard for Distributed Interactive Simulation - Application Protocols*, páginas 337–340. Estados Unidos, 2012. 30

[84] *Protobuf documentation: Overview*. https://protobuf.dev/overview/#solve, acesso em 31/05/2024. 30

[85] Densmore, James: *Data Pipelines Pocket Reference: Moving and Processing Data for Analytics*, páginas 21–29. O'Reilly, 2021. 31

[86] Steen, Maarten van e Andrew S. Tanenbaum: *Distributed Systems*, páginas 33–34. Maarten van Steen, quarta edição, 2024. 32

[87] Bani, Fajar Ciputra Daeng, Suharjito, Diana e Abba Suganda Girsang: *Implementation of database massively parallel processing system to build scalability on process data warehouse*. Procedia Computer Science, 135:69–70, 2018. https://doi.org/10.1016/j.procs.2018.08.151. 32

[88] Marz, Nathan e James Warren: *Big Data: Principles and best practices of scalable real-time data systems*, páginas 7–8. Manning Publications Co., 2015. 32

[89] Simon, Alan: *Data Lakes For Dummies*, páginas 39–42. Wiley & Sons, 2021. 35

[90] Skyrius, Rimvydas: *Business Intelligence: A Compreensive Approach to Information Needs, Technologies and Culture*, páginas 163–164. Springer, 2021. https://doi.org/10.1007/978-3-030-67032-0. 35

[91] Defense Science Board: *Summer Study on Autonomy*, página 5. 2016. https://apps.dtic.mil/sti/citations/AD1017790. 35

[92] Russel, Stuart e Peter Norvig: *Artificial Intelligence: A Modern Approach*, página 2. Pearson, quarta edição, 2021. 35

[93] Källström, Johan e Fredrik Heintz: *Reinforcement learning for computer generated forces using open-source software*. 2019. https://www.xcdsystem.com/iitsec/proceedings/index.cfm?Year=2019&AbID=27519&CID=48#View. 35

[94] Etheredge, Charles, Kyle Russell, Willian Marx, Timothy Hill e Daron Drown: *The use of ai/ml to replicate threat behaviors for nonlinear simulation*. 2022. https://www.xcdsystem.com/iitsec/proceedings/index.cfm?Year=2022&AbID=112369&CID=944#View. 35

[95] Vatral, Caleb, Gautam Biswas, Naveeduddin Mohammed e Benjamin S. Goldberg: *Automated assessment of team performance using multimodal bayesian learning analytics*. 2022. https://www.xcdsystem.com/iitsec/proceedings/index.cfm?Year=2022&AbID=112414&CID=944#View. 35

[96] Simon, Alan: *Data Lakes For Dummies*, página 61. Wiley & Sons, 2021. 37

[97] Richards, Mark e Neal Ford: *Fundamentals of Software Architecture: An Engineering Approach*, páginas 3–7. O'Reilly, 2020. 37

[98] Bass, Len, Paul Clements e Rick Kazman: *Software Architecture in Practice*, páginas 3–6. Pearson, terceira edição, 2013. 37

[99] Cervantes, Humberto e Rick Kazman: *Designing Software Architectures: A Practical Approach*, capítulo 2. Person, 2016. 37

[100] Richards, Mark e Neal Ford: *Fundamentals of Software Architecture: An Engineering Approach*, páginas 58–64. O'Reilly, 2020. 38

[101] ISO/IEC 25010: *ISO/IEC 25010:2011, systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models*, 2011. 38

[102] Kraus, Johann M., Ludwig Lausser, Franz Jobst, Michaela Bock, Carolin Halanke, Michael Hummel, Peter Heuschmann e Hans A. Kestler: *Big data and precision medicine: challenges and strategies with health care data.* International Journal of Data Science and Analytics, 6:241–249, 2018. https://doi.org/10.1007/s41060-018-0095-0. 39

[103] Prasser, Fabian, Oliver Kohlbacher, Ulrich Mansmann, Bernhard Bauer e Klaus A. Kuhn: *Data integration for future medicine (difuture). an architectural and methodological overview.* Methods of Information in Medicine, 57(01):e57–e65, 2018. https://doi.org/10.3414/ME17-02-0022. 40, 43

[104] Winter, Alfred, Sebastian Staubert, Danny Ammon, Stephan Aiche, Oya Beyan, Verena Bischoff, Philipp Daumke, Stefan Decker, Gert Funkat, Jan E. Gewehr, Armin de Greiff, Silke Haferkamp, Udo Hahn, Andreas Henke, Toralf Kirsten, Thomas Klöss, Jörg Lippert, Matthias Löbe, Volker Lowitsch, Oliver Maassen, Jens Maschmann, Sven Meister, Rafael Mikolajczyk, Matthias Nüchter, Mathias W. Pletz, Erhard Rahm, Morris Riedel, Kutaiba Saleh, Andreas Schuppert, Stefan Smers, André Stollenwerk, Stefan Uhlig, Thomas Wendt, Sven Zenker, Wolfgang Fleig, Gernot Marx, André Scherag e Markus Löffler: *Smart medical information technology for healthcare (smith). data integration based on interoperability standards.* Methods of Information in Medicine, 57(01):e92–e105, 2018. https://doi.org/10.3414/ME18-02-0004. 40, 43

[105] Štufi, Martin, Boris Bačić e Leonid Stoimenov: *Big data analytics and processing platform in czech republic healthcare.* Applied Sciences, 10(5):1–23, 2020. https://doi.org/10.3390/app10051705. 40, 43

[106] *Talend*. https://www.talend.com/, acesso em 2024/09/10. 41

[107] *Vertica*. https://www.vertica.com/, acesso em 2024/09/10. 41

[108] *Tableau*. https://www.tableau.com/pt-br, acesso em 2024/09/10. 41

[109] Wang, Miye, Sheyu Li, Tao Zheng, Qingke Shi, Xuejun Zhuo, Renxin Ding e Yong Huang: *Big data health care platform with multisource heterogeneous data integration and massive high-dimensional data governance for large hospitals: Design, development, and application.* JMIR Medical Informatics, 10(4):1–15, 2022. https://doi.org/10.2196/36481. 41, 42, 43

[110] Parciak, Marcel, Markus Suhr, Chirstian Schmidt, Caroline Bönisch, Benjamin Löhnardt, Dorotea Kesztyüs e Tibor Kesztyüs: *Fairness through automation: Development of an automated medical data integration infraestructure for fair health data in*

a maximum care university hospital. BMC Medical Informatics and Decision Making, 23(94):1–14, 2023. https://doi.org/10.1186/s12911-023-02195-3. 41, 43

[111] Hoffmann, Katja, Anne Pelz, Elena Karg, Andrea Gottschalk, Thomas Zerjatke, Silvio Schuster, Heiko Böhme, Ingmar Glauche e Ingo Roeder: *Data integration between clinical research and patient care: A framework for context-depending data sharing and in silico predictions*. PLOS Digital Health, 2(5):1–17, 2023. https://doi.org/10.1371/journal.pdig.0000140. 41, 43

[112] Lu, Fengshun, Xingzhi Hu, Bendong Zhao, Xiong Jiang, Duoneng Liu, Jianqi Lai e Zhiren Wang: *Review of the research progress in combat simulation software*. Applied Sciences, 13(9), 2023. https://doi.org/10.3390/app13095571. 43, 45

[113] King, David W., Douglas D. Hodson e Gilbert L. Peterson: *The role of simulation frameworks in relation to experiments*. páginas 4153–4174, 2017. https://doi.org/10.1109/WSC.2017.8248123. 44, 45

[114] Clive, Peter D., Jeffrey A. Johnson, Michael J. Moss, James M. Zeh, Britain M. Birkmire e Douglas D. Hodson: *Advanced framework for simulation, integration and modeling (afsim)*. International Conference of Scientific Computing, páginas 73–77, 2015. 44, 45

[115] West, Timothy D. e Brian Birkmire: *Afsim: The air force research laboratory's approach to making m&s ubiquitous in the weapon system concept development process*. Journal of Cyber Security and Information Systems, 7(3):50–55, 2019. 44, 45

[116] Dantas, Joao P. A, Andre N. Costa, Vitor C. F. Gomes, Andre R. Kuroswiski, Felipe L. L. Medeiros e Diego Geraldo: *Asa: A simulation environment for evaluating military operational scenarios*. Em *Proceedings of the 20th International Conference on Scientific Computing*, páginas 25–28, Las Vegas, Estados Unidos, 2022. 44, 45

[117] Dantas, Joao P. A., Diego Geraldo, Andre N. Costa, Marcos R. O. A. Maximo e Takashi Yoneyama: *Asa-simaas: Advancing digital transformation through simulation services in the brazilian air force*. Em *Simpósio de Aplicações Operacionais em Áreas de Defesa 2023 (SIGE2023)*, São José dos Campos, SP, 2023. 44, 45

[118] *Mixr: The mixed reality simulation platform*. https://www.mixr.dev, acesso em 2024/09/10. 44

[119] *Unreal engine option*. https://flamesframework.com/products/flames-options/unreal-engine-option-flames/. 45

[120] Ternion Corporation: *Overview of the flames simulation framework*. Relatório Técnico. https://flamesframework.com/flames-overview. 45

[121] Etzkowitz, Henry e Loet Leydesdorff: *The dynamics of innovation: from national systems and "mode 2" to a triple helix of university–industry–government relations*. Research Policy, 29(2):109–123, 2000. 10.1016/S0048-7333(99)00055-4. 46

[122] Steen, Maarten van e Andrew S. Tanenbaum: *Distributed Systems*, páginas 56–68. Maarten van Steen, quarta edição, 2024. 52

[123] Fielding, Roy T. e Julian Reschke: *Hypertext transfer protocol (http/1.1): Message syntax and routing*. RFC 7230, 2014. https://www.rfc-editor.org/rfc/rfc7230. 54

[124] *Keycloak*. https://www.keycloak.org, acesso em 2025/06/10. 57

[125] *How openid connect works - openid foundation*. https://openid.net/developers/how-connect-works, acesso em 2025/06/10. 57

[126] *Single sign-on*. https://auth0.com/docs/authenticate/single-sign-on, acesso em 2025/06/10. 57

[127] *keycloak/keycloak - docker image | docker hub*. https://hub.docker.com/r/keycloak/keycloak, acesso em 2025/06/10. 57

[128] *Single sign-on with keycloak | nginx documentation*. https://docs.nginx.com/nginx/deployment-guides/single-sign-on/keycloak, acesso em 2025/06/10. 58

[129] *Authentication | wiki.js*. https://docs.requarks.io/auth, acesso em 2025/06/10. 58

[130] *A guide for configuring keycloak as a authentication provider in wiki-js | https://wiki.js.org | feature request for adding this to the docs: https://requarks.canny.io/wiki/p/keycloak-auth-docs-proposal-for-a-guide-written*. https://gist.github.com/Sherex/283d1e4ef07b2bf0a930417dc0117238, acesso em 2025/06/10. 58

[131] *Fab auth manager authentication — apache-airflow-providers-fab documentation*. https://airflow.apache.org/docs/apache-airflow-providers-fab/stable/auth-manager/webserver-authentication.html#example-using-team-based-authorization-with-keycloak, acesso em 2025/06/11. 58

[132] *Openid connect access management | aistor object store documentation*. https://docs.min.io/enterprise/aistor-object-store/administration/iam/access/oidc-access, acesso em 2025/06/11. 58

[133] *Integrate minio with keycloak oidc*. https://blog.min.io/integrate-minio-with-keycloak-oidc, acesso em 2025/06/11. 58

[134] *Configuring keycloak as an sso – dremio support*. https://support.dremio.com/hc/en-us/articles/29109477930651-Configuring-KeyCloak-as-an-SSO, acesso em 2025/06/11. 58

[135] *Jupyterhub and oauth — jupyterhub documentation*. https://jupyterhub.readthedocs.io/en/latest/explanation/oauth.html#jupyterhub-and-oauth, acesso em 2025/06/11. 58

[136] *Configuring superset | superset.* https://superset.apache.org/docs/configuration/configuring-superset/#keycloak-specific-configuration-using-flask-oidc, acesso em 2025/06/11. 58

[137] *nginx.* https://nginx.org/en, acesso em 2025/06/11. 58

[138] *nginx - official image | docker hub.* https://hub.docker.com/_/nginx, acesso em 2025/06/11. 58

[139] *Wiki.js.* https://js.wiki, acesso em 2025/06/11. 58

[140] *linuxserver/wikijs - docker image | docker hub.* https://hub.docker.com/r/linuxserver/wikijs, acesso em 2025/06/11. 58

[141] Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg *et al.*: *The fair guiding principles for scientific data management and stewardship.* Scientific Data, 3(1), 2016. 58

[142] *Sobre a node.js.* https://nodejs.org/pt/about, acesso em 2025/06/11. 59, 60

[143] *node - official image | docker hub.* https://hub.docker.com/_/node, acesso em 2025/06/11. 59, 61

[144] *What is airflow®? — airflow 3.0.0 documentation.* https://airflow.apache.org/docs/apache-airflow/stable/index.html, acesso em 2025/06/12. 59

[145] Harenslak, Bes e Julian de Ruiter: *Data Pipelines with Apache Ariflow*, páginas 3–13. Manning Publications, 2021. 59

[146] *Postgresql: About.* https://www.postgresql.org/about, acesso em 2025/06/12. 60

[147] *apache/airflow - docker image | docker hub.* https://hub.docker.com/r/apache/airflow, acesso em 2025/06/12. 60

[148] *protobufjs/protobuf.js: Protocol buffers for javascript & typescript.* https://github.com/protobufjs/protobuf.js, acesso em 2025/06/11. 61, 70

[149] Fette, Ian e Alexey Melnikov: *The websocket protocol.* RFC 6455, 2011. https://www.rfc-editor.org/rfc/rfc6455. 61

[150] *Data lakehouse solutions | minio.* https://min.io/solutions/modern-data-lakes-lakehouses, acesso em 2025/06/12. 61

[151] *Overview | parquet.* https://parquet.apache.org/docs/overview, acesso em 2025/06/12. 65

[152] *Documentation | apache avro.* https://avro.apache.org/docs, acesso em 2025/06/12. 65

[153] *minio/minio - docker image | docker hub.* https://hub.docker.com/r/minio/minio, acesso em 2025/06/12. 66

[154] *Apache spark - unified engine for large-scale data analytics.* https://spark.apache.org, acesso em 2025/06/12. 67

[155] Chambers, Bill e Matei Zaharia: *Spark: The Definitive Guide*, página 12. O' Reilly, 2018. 67

[156] *open-dis/open-dis-python: Python implementation of the ieee-1278.1 distributed interactive simulation (dis) application protocol v7.* https://github.com/open-dis/open-dis-python, acesso em 2025/06/12. 68

[157] *Python generated code guide | protocol buffers documentation.* https://protobuf.dev/reference/python/python-generated, acesso em 2025/06/12. 68

[158] *apache/spark - docker image | docker hub.* https://hub.docker.com/r/apache/spark, acesso em 2025/06/12. 68

[159] *Why dremio | benefits of working with us | dremio.* https://www.dremio.com/why-dremio, acesso em 2025/06/12. 68

[160] *Project nessie: Transactional catalog for data lakes with git-like semantics.* https://projectnessie.org, acesso em 2025/06/12. 68

[161] *Apache iceberg - apache iceberg™.* https://iceberg.apache.org, acesso em 2025/06/12. 68

[162] Shiran, Tomer, Jason Hughes e Alex Merced: *Apache Iceberg: The Definitive Guide: Data Lakehouse Functionality, Performance, and Scalability on the Data Lake*, página 145. O' Reilly, 2024. 68

[163] *Dremio open source: Explore nessie | dremio.* https://www.dremio.com/open-source/nessie, acesso em 2025/06/12. 69

[164] *A developer's introduction to apache iceberg using minio.* https://blog.min.io/a-developers-introduction-to-apache-iceberg-using-minio/, acesso em 2025/06/12. 69

[165] *dremio/dremio-oss - docker image | docker hub.* https://hub.docker.com/r/dremio/dremio-oss, acesso em 2025/06/12. 69

[166] *projectnessie/nessie - docker image | docker hub.* https://hub.docker.com/r/projectnessie/nessie, acesso em 2025/06/12. 69

[167] *Project jupyter | about us.* https://jupyter.org/about, acesso em 2025/06/12. 69

[168] *jupyterhub/jupyterhub - docker image | docker hub.* https://hub.docker.com/r/jupyterhub/jupyterhub, acesso em 2025/06/12. 69

[169] *Welcome | superset.* https://superset.apache.org, acesso em 2025/06/12. 69

[170] *Imagem docker do superset.* https://hub.docker.com/r/apache/superset, acesso em 2025/06/12. 70

[171] MASA: *MASA Sword Network API Documentation*, páginas 5–9. 70

[172] *Docker desktop.* https://docs.docker.com/desktop/, acesso em 2025/06/15. 71

[173] *Docker desktop wsl 2 backend on windows.* https://docs.docker.com/desktop/features/wsl/, acesso em 2025/06/15. 71

[174] *Kubernetes.* https://kubernetes.io/pt-br, acesso em 2025/06/15. 71

[175] *Running airflow in docker — airflow 3.0.4 documentation.* https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html#running-airflow-in-docker, acesso em 2025/06/15. 71

[176] *Docker compose | superset.* https://superset.apache.org/docs/installation/docker-compose, acesso em 2025/06/15. 71

[177] *Jupyterhub — jupyterhub documentation.* https://jupyterhub.readthedocs.io/en/latest/#distributions, acesso em 2025/06/15. 71

[178] *Danovacavalaria/simulation-data-integration: Implementation code for the thesis "architecture for integration of combat simulation data within the brazilian army land force".* https://github.com/DANOVACAVALARIA/simulation-data-integration, acesso em 2025/06/15. 71

[179] Fenton, Norman e James Bieman: *Sotware Metrics: A Rigorous and Practical Approach.* CRC Press, terceira edição, 2015. 72

[180] *Benchmarking methodology for network interconnect devices (rfc 2544) (1999).* https://www.rfc-editor.org/rfc/rfc2544, acesso em last accessed 2025/05/16. 74

Apêndice A

Fichamento de Artigo Científico