

Pedro Miguel de Paula Silva

ThreatLens: Desenvolvimento de um *dashboard* para análise de dados

Uberlândia - MG

2026

Pedro Miguel de Paula Silva

ThreatLens: Desenvolvimento de um *dashboard* para análise de dados

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Universidade Federal de Uberlândia

Faculdade de Computação

Bacharelado em Sistemas de Informação

Orientador: Prof. Dr. Paulo Henrique Ribeiro Gabriel

Uberlândia - MG

2026

Agradecimentos

Primeiramente, agradeço a Deus, por Sua presença constante em minha vida, por me conceder sabedoria nos momentos de dificuldade, coragem para superar os desafios e por iluminar meus caminhos até a realização deste sonho. Sem a Sua graça e misericórdia, esta conquista não seria possível.

Aos meus familiares, expresso minha eterna gratidão pelo amor e incentivo durante toda a minha trajetória acadêmica. Em especial, agradeço ao meu pai, Marcelo, e à minha mãe, Albiana, por acreditarem em mim, por todo o esforço e por estarem ao meu lado em cada momento dessa caminhada. Esta vitória também é de vocês.

Ao meu orientador, Prof. Dr. Paulo Henrique Ribeiro Gabriel, agradeço pela dedicação, paciência e pelos conhecimentos compartilhados durante o desenvolvimento deste trabalho. Suas orientações foram essenciais para a realização desta pesquisa e para o meu crescimento acadêmico e profissional.

Aos professores que fizeram parte da minha formação, agradeço por todos os ensinamentos, pela dedicação ao ensino e pelas contribuições que foram fundamentais para minha trajetória universitária. Cada conhecimento compartilhado contribuiu para a construção do profissional que estou me tornando.

Por fim, agradeço à Anna Clara, minha querida amiga que a universidade me presenteou e parceira no desenvolvimento do ThreatLens, responsável pelo *back-end* da plataforma, por compartilhar comigo os desafios e aprendizados ao longo dessa jornada. Foi um privilégio poder construir este projeto ao seu lado e compartilhar esta conquista tão especial.

Resumo

Este trabalho apresenta o desenvolvimento do *front-end* da plataforma ThreatLens, uma aplicação web voltada à visualização e análise de dados provenientes de um sistema de detecção de ameaças cibernéticas. Concebido como a camada de apresentação visual de um projeto de pesquisa mais amplo em Inteligência de Ameaças Cibernéticas (*Cyber Threat Intelligence* – CTI), o sistema consolida e organiza informações coletadas e classificadas por iniciativas complementares. Para a construção da interface, foi utilizada a biblioteca JavaScript React, em conjunto com o React Router DOM para o gerenciamento de rotas, CSS para estilização responsiva e a biblioteca Chart.js para a visualização gráfica dos dados. A comunicação com o servidor foi realizada por meio de requisições HTTP assíncronas às APIs RESTful disponibilizadas pelo *back-end*. O desenvolvimento contemplou o levantamento de requisitos funcionais e não funcionais, a prototipação de alta fidelidade no Figma e a implementação de componentes, páginas e serviços organizados segundo uma arquitetura modular baseada em separação de responsabilidades. Como resultado, obteve-se uma plataforma responsiva, interativa e integrada ao *back-end*, capaz de apresentar indicadores, gráficos e publicações classificadas de forma clara e acessível, contribuindo para que profissionais de cibersegurança possam interpretar o cenário de ameaças e tomar decisões proativas.

Palavras-chave: Inteligência de Ameaças Cibernéticas. *Dashboard*. React. Cibersegurança. Visualização de dados.

Abstract

This work presents the development of the front-end of the ThreatLens platform, a web application aimed at visualizing and analyzing data produced by a cyber threat detection system. Conceived as the visual presentation layer of a broader Cyber Threat Intelligence (CTI) research project, the system consolidates and organizes information collected and classified by complementary back-end development initiatives. The interface was built using the React JavaScript library, together with React Router DOM for route management, CSS for responsive styling, and the Chart.js library for graphical data visualization. Communication with the server was carried out through asynchronous HTTP requests to the RESTful APIs provided by the back-end. The development process included gathering functional and non-functional requirements, high-fidelity prototyping in Figma, and the implementation of components, pages, and services organized according to a modular architecture based on separation of concerns. As a result, a responsive and interactive platform was obtained, integrated with the back-end and capable of presenting indicators, charts, and classified posts in a clear and accessible manner, helping cybersecurity professionals interpret the threat landscape and make proactive decisions.

Keywords: Cyber Threat Intelligence. Dashboard. React. Cybersecurity. Data visualization.

Lista de ilustrações

Figura 1 – Protótipo de alta fidelidade do <i>dashboard</i> desenvolvido no Figma	34
Figura 2 – Tela de login do ThreatLens	38
Figura 3 – Tela de verificação de código	39
Figura 4 – <i>Dashboard</i> do ThreatLens	40
Figura 5 – Nuvem de palavras do <i>dashboard</i> do ThreatLens	40
Figura 6 – Tela de publicações analisadas	41
Figura 7 – Tela Minha conta	42
Figura 8 – Tela administrativa	43
Figura 9 – Filtragem ativa de publicações por meio de busca textual	47
Figura 10 – Publicações ordenadas pela classificação em ordem decrescente	48
Figura 11 – Publicações ordenadas pela classificação em ordem crescente	48
Figura 12 – Mensagem de erro de autenticação por e-mail ou senha incorretos . . .	49
Figura 13 – Indicadores e publicações mais relevantes do <i>dashboard</i> em dispositivo móvel	50
Figura 14 – Gráfico de relevância das publicações em dispositivo móvel	51
Figura 15 – Gráfico das fontes de CTI e nuvem de palavras em dispositivo móvel .	52
Figura 16 – Plataforma com o tema escuro (<i>Dark Mode</i>) ativado	53

Lista de Quadros

Quadro 1 – RF01: Cadastro de usuário	30
Quadro 2 – RF02: Autenticação	30
Quadro 3 – RF03: Redefinição de senha	30
Quadro 4 – RF04: Logout	30
Quadro 5 – RF05: Consulta de perfil e alteração de senha	31
Quadro 6 – RF06: Carregamento de publicações	31
Quadro 7 – RF07: Busca de publicações	31
Quadro 8 – RF08: Ordenação de publicações	31
Quadro 9 – RF09: Visualização de <i>dashboard</i>	31
Quadro 10 – RF10: Navegação entre páginas	32
Quadro 11 – RF11: Exibição de mensagens de feedback	32
Quadro 12 – RF12: Remoção de contas	32
Quadro 13 – RF13: Alternância de tema	32
Quadro 14 – RNF01: Responsividade	32
Quadro 15 – RNF02: Segurança	33
Quadro 16 – RNF03: Usabilidade	33
Quadro 17 – RNF04: Manutenibilidade	33

Lista de abreviaturas e siglas

API	Interface de Programação de Aplicação (<i>Application Programming Interface</i>)
APT	Ameaça Persistente Avançada (<i>Advanced Persistent Threat</i>)
BYOD	Traga Seu Próprio Dispositivo (<i>Bring Your Own Device</i>)
CLI	Interface de Linha de Comando (<i>Command Line Interface</i>)
CSS	Folhas de Estilo em Cascata (<i>Cascading Style Sheets</i>)
CTI	Inteligência de Ameaças Cibernéticas (<i>Cyber Threat Intelligence</i>)
DBSCAN	Agrupamento Espacial Baseado em Densidade de Aplicações com Ruído (<i>Density-Based Spatial Clustering of Applications with Noise</i>)
DDoS	Negação de Serviço Distribuída (<i>Distributed Denial of Service</i>)
DOM	Modelo de Objetos do Documento (<i>Document Object Model</i>)
FACOM	Faculdade de Computação
HMR	<i>Hot Module Replacement</i>
HTML	Linguagem de Marcação de Hipertexto (<i>HyperText Markup Language</i>)
HTTP	Protocolo de Transferência de Hipertexto (<i>Hypertext Transfer Protocol</i>)
IA	Inteligência Artificial
IDE	Ambiente de Desenvolvimento Integrado (<i>Integrated Development Environment</i>)
IoC	Indicador de Comprometimento (<i>Indicator of Compromise</i>)
IoT	Internet das Coisas (<i>Internet of Things</i>)
IP	Protocolo de Internet (<i>Internet Protocol</i>)
KNN	K Vizinhos Mais Próximos (<i>K-Nearest Neighbors</i>)
LDA	Alocação Latente de Dirichlet (<i>Latent Dirichlet Allocation</i>)
PLN	Processamento de Linguagem Natural

REST	Transferência de Estado Representacional (<i>Representational State Transfer</i>)
RF	Requisito Funcional
RNF	Requisito Não Funcional
SEO	Otimização para Mecanismos de Busca (<i>Search Engine Optimization</i>)
SOAR	Orquestração, Automação e Resposta de Segurança (<i>Security Orchestration, Automation and Response</i>)
SPA	Aplicação de Página Única (<i>Single Page Application</i>)
TF-IDF	Frequência do Termo–Frequência Inversa do Documento (<i>Term Frequency–Inverse Document Frequency</i>)
TIP	Plataforma de Inteligência de Ameaças (<i>Threat Intelligence Platform</i>)
TOR	<i>The Onion Router</i>
TTP	Táticas, Técnicas e Procedimentos (<i>Tactics, Techniques, and Procedures</i>)
UFU	Universidade Federal de Uberlândia
URL	Localizador Uniforme de Recursos (<i>Uniform Resource Locator</i>)

Sumário

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	Ameaça Cibernética	14
2.2	Cibersegurança	15
2.3	Inteligência de Ameaças Cibernéticas	16
2.4	<i>Surface Web, Deep Web e Dark Web</i>	17
3	TRABALHOS RELACIONADOS	20
4	THREATLENS: PLATAFORMA PARA ANÁLISE DE DADOS . . .	23
4.1	Visão geral da plataforma	23
4.2	Componentes do sistema	24
4.3	Público-alvo e finalidade	24
5	TECNOLOGIAS UTILIZADAS	25
5.1	Figma	25
5.2	React	25
5.3	React Router DOM	26
5.4	CSS	26
5.5	Chart.js	27
5.6	Git e GitHub	27
5.7	VS Code	28
5.8	Vite	28
6	DESENVOLVIMENTO	29
6.1	Levantamento de requisitos	29
6.1.1	Requisitos Funcionais	30
6.1.2	Requisitos Não Funcionais	32
6.2	Prototipação de alta fidelidade	33
6.3	Etapas da implementação	35
6.3.1	Criação do projeto	35
6.3.2	Estrutura de diretórios	35
6.3.3	Desenvolvimento de componentes	36
6.3.4	Desenvolvimento de páginas	37
6.3.5	Integração com o <i>back-end</i>	43

7	AVALIAÇÃO	46
7.1	Procedimentos de avaliação	46
7.2	Avaliação do fluxo de uso e requisitos	46
7.2.1	Acesso e Autenticação	46
7.2.2	Visão Geral (<i>Dashboard</i>)	46
7.2.3	Consulta de Publicações	47
7.2.4	Gerenciamento de Contas	49
7.2.5	<i>Feedback</i> Visual	49
7.2.6	Responsividade e Alternância de Tema	49
7.2.7	Manutenibilidade	53
7.3	Resultados da avaliação	53
8	CONCLUSÃO	54
	APÊNDICE A – DECLARAÇÃO DE USO DE IA GENERATIVA	55
	REFERÊNCIAS	56

1 Introdução

A Internet é uma rede de computadores que conecta milhares de dispositivos ao redor do mundo, tendo começado a se desenvolver na década de 1960 a partir de projetos acadêmicos financiados pelo governo dos Estados Unidos, com a criação da ARPANET, uma rede para comunicação entre universidades e laboratórios de pesquisa (KUROSE; ROSS, 2013). Inicialmente restrita a pesquisadores, a internet evoluiu rapidamente, alcançando cerca de 5,56 bilhões de usuários em 2025 (DATAREPORTAL, 2025) e tornando-se uma ferramenta essencial no cotidiano. Contudo, essa massificação trouxe um aumento significativo nas ameaças cibernéticas, que se tornaram mais sofisticadas e organizadas, expondo usuários e sistemas a riscos crescentes de segurança e privacidade.

Em 2021, o ataque ocorrido à empresa norte-americana Colonial Pipeline resultou na paralisação de um dos sistemas de abastecimento de combustível dos Estados Unidos, provocando escassez e prejuízos financeiros. Segundo estimativas do relatório elaborado pela Cybersecurity Ventures, os crimes cibernéticos geraram perdas globais da ordem de 8 trilhões de dólares apenas no ano de 2023 (CYBERSECURITY VENTURES, 2023).

Nesse contexto, cresce o interesse de pesquisadores e especialistas em segurança digital por uma nova abordagem voltada à proteção contra ameaças cibernéticas: a Inteligência de Ameaças Cibernéticas (*Cyber Threat Intelligence* – CTI). Segundo McMillan (2013), essa prática envolve a obtenção e a análise de dados fundamentados em evidências sobre ameaças em potencial, incluindo aspectos como contexto, métodos utilizados pelos atacantes, indicadores de comprometimento, possíveis impactos e orientações práticas para auxiliar na definição de medidas de resposta adequadas. A partir das informações obtidas, as organizações são capazes de tomar decisões proativas e eficazes em relação à segurança, abrangendo a detecção, a prevenção e a recuperação de ataques cibernéticos.

Diversos trabalhos vêm demonstrando a importância da CTI no fortalecimento da segurança cibernética. Um exemplo é a pesquisa desenvolvida por Jesus Filho (2023), que propôs um modelo voltado para a identificação de posts maliciosos na *Dark Web* por meio de aprendizado de máquina e processamento de linguagem natural. Esse modelo realiza a coleta automatizada de conteúdos publicados em fóruns da *Dark Web* e os classifica com base em indicadores de comprometimento (IoCs), palavras-chave contextuais e análise manual, permitindo filtrar e priorizar informações relevantes para o contexto de cibersegurança.

Em continuidade a essa iniciativa, este trabalho propõe o desenvolvimento da plataforma web ThreatLens, concebida como a camada de apresentação visual de um projeto de pesquisa mais amplo voltado à inteligência de ameaças cibernéticas. Desenvolvido em

cooperação com iniciativas complementares de desenvolvimento *back-end*, o sistema consolida e organiza os dados coletados e classificados a partir das metodologias propostas em pesquisas anteriores. O objetivo geral deste trabalho é disponibilizar uma plataforma para visualização e análise de dados. Para isso, são estabelecidos como objetivos específicos: realizar o levantamento de requisitos funcionais e não funcionais; criar a prototipação de alta fidelidade da interface; implementar páginas, componentes e serviços organizados de forma modular; e integrar a plataforma com o *back-end* por meio de uma API para o consumo dos dados. Dessa forma, busca-se disponibilizar uma interface gráfica intuitiva, responsiva e interativa que auxilie profissionais de cibersegurança na análise do cenário de ameaças, oferecendo informações estruturadas de forma ágil para a tomada de decisões proativas e embasadas.

Para o desenvolvimento da plataforma proposta, foi utilizada a biblioteca JavaScript React, amplamente reconhecida por sua eficiência na criação de interfaces dinâmicas, interativas e baseadas em componentes reutilizáveis. Com o objetivo de proporcionar uma experiência de usuário fluida e adaptável a diferentes dispositivos, foram aplicadas práticas de design responsivo por meio da linguagem de estilização CSS. A comunicação com o servidor ocorre por meio de requisições HTTP assíncronas, utilizando a biblioteca Axios, para o consumo das APIs RESTful previamente implementadas. Para a visualização dos dados coletados e classificados, foi empregada a biblioteca gráfica Chart.js, que possibilita a exibição de gráficos e indicadores visuais relacionados às ameaças cibernéticas, com base nas informações retornadas pelas chamadas à API. Por fim, para o controle de versionamento do código-fonte, foram utilizadas a ferramenta Git e a plataforma GitHub, possibilitando o gerenciamento eficiente do histórico de alterações e a manutenção organizada do projeto ao longo do seu desenvolvimento.

O estilo de maturidade de pesquisa adotado neste trabalho é o Tecnológico, conforme a classificação proposta por [Wazlawick \(2020\)](#). Esta escolha se justifica pelo caráter prático e aplicado do trabalho, que visa o desenvolvimento de uma plataforma funcional para análise de dados provenientes de um sistema de detecção de ameaças cibernéticas. A proposta não se restringe à investigação teórica ou à validação de hipóteses, mas concentra-se na construção, integração e implementação de uma solução computacional concreta. O sistema desenvolvido consolida avanços obtidos em pesquisas anteriores e propõe melhorias operacionais e de usabilidade que possam ser efetivamente utilizadas por profissionais de cibersegurança.

Este trabalho está organizado em oito capítulos. O Capítulo 2 reúne a fundamentação teórica sobre ameaças cibernéticas, cibersegurança, CTI e as camadas da Web. O Capítulo 3 discute as pesquisas que embasam o projeto e as propostas relacionadas à visualização de dados de segurança. O Capítulo 4 apresenta a plataforma ThreatLens, sua visão geral, seus componentes, seu público-alvo e sua finalidade. O Capítulo 5 des-

creve as tecnologias utilizadas no desenvolvimento do sistema. O Capítulo 6 detalha o levantamento de requisitos, a prototipação de alta fidelidade, a arquitetura da aplicação e a implementação da interface, incluindo sua integração com o *back-end*. O Capítulo 7 é dedicado à avaliação da plataforma. Por fim, o Capítulo 8 apresenta as conclusões, as principais contribuições e as possibilidades de trabalhos futuros.

2 Fundamentação Teórica

Este capítulo reúne os fundamentos teóricos necessários ao desenvolvimento do trabalho, contemplando desde os principais tipos de ameaças cibernéticas e os princípios da cibersegurança até a Inteligência de Ameaças Cibernéticas (CTI) em seus diferentes níveis de aplicação. Também é abordada a estrutura em camadas da Internet, contemplando a *Surface Web*, *Deep Web* e *Dark Web* e seus impactos na segurança da informação.

2.1 Ameaça Cibernética

Uma ameaça cibernética é qualquer situação que possa colocar em risco a segurança da informação, explorando falhas ou vulnerabilidades para causar prejuízos financeiros, operacionais ou à reputação de uma pessoa ou organização (STALLINGS; BROWN, 2012). Em termos simples, trata-se de qualquer ação mal-intencionada capaz de comprometer sistemas, dados ou serviços digitais.

Segundo a IBM (2024b), existem diferentes tipos de ameaças, cada uma com métodos e objetivos específicos. Entre as mais comuns, destacam-se:

- **Malware:** é o termo genérico para programas criados com a intenção de prejudicar sistemas ou roubar informações. Inclui vírus (que infectam arquivos), *worms* (que se espalham automaticamente pela rede), *trojans* (que se disfarçam de programas legítimos) e *ransomware* (que bloqueia dados e exige pagamento para liberá-los). Um exemplo famoso foi o *ransomware* WannaCry (2017), que afetou hospitais e empresas no mundo inteiro.
- **Phishing:** técnica de enganar pessoas para obter dados confidenciais, como senhas ou informações bancárias. Normalmente é feita por e-mails ou mensagens falsas que imitam empresas conhecidas.
- **Ataques de Negação de Serviço Distribuída (DDoS):** sobrecarregam servidores com tantas solicitações que eles deixam de funcionar para usuários legítimos. Um caso marcante foi o ataque ao provedor Dyn (2016), que interrompeu temporariamente serviços como Twitter, Netflix e Spotify.
- **Ameaças Persistentes Avançadas (APTs):** ataques planejados e de longa duração, geralmente conduzidos por grupos organizados ou patrocinados por governos. O objetivo é entrar discretamente em redes corporativas ou governamentais para roubar dados estratégicos, sem ser detectado por longos períodos.

- **Exploração de Vulnerabilidades *Zero-Day*:** ataques que aproveitam falhas desconhecidas pelos fabricantes e ainda sem correção disponível. São perigosos porque, no início, não existe defesa específica contra eles.

A forma como essas ameaças surgem e evoluem mudou ao longo do tempo. Nos primeiros anos da computação, ações maliciosas eram mais experimentais e, muitas vezes, feitas por curiosidade, geralmente em sistemas isolados, o que tornava a contenção mais simples. No entanto, com a popularização da Internet no final da década de 1980, começaram a aparecer os primeiros vírus e *worms* em larga escala. Nos anos 1990, o cenário se diversificou, com o crescimento do *spam*, dos ataques DDoS e de fraudes com cartões de crédito. Já nos anos 2000, surgiram mercados clandestinos especializados, onde criminosos vendiam credenciais roubadas, desenvolviam *malwares* profissionalmente e aplicavam esquemas de *cash-out* para converter dados roubados em dinheiro (ANDERSON, 2020).

Mais recentemente, os ataques se tornaram mais sofisticados com o uso de tecnologias como Inteligência Artificial (IA) e computação em nuvem. Essas ferramentas permitem criar campanhas automáticas, atingir um número muito maior de alvos e desenvolver golpes mais convincentes, como o uso de *deepfakes* para enganar vítimas (MICROSOFT, 2024).

Diante desse cenário, é fundamental adotar estratégias proativas de defesa, investir em inteligência de ameaças e promover programas de conscientização em segurança digital. Essas medidas reduzem riscos e ajudam a proteger dados e sistemas contra um ambiente cibernético cada vez mais complexo.

2.2 Cibersegurança

A cibersegurança envolve reunir e alinhar recursos, como profissionais, infraestrutura, estruturas organizacionais e processos, de modo a proteger redes e sistemas computacionais conectados contra incidentes que possam violar sua integridade e prejudicar direitos de propriedade, causando algum nível de dano ou perda (SCHILIRO, 2022). Esse conceito surgiu na década de 1970 com a criação dos primeiros sistemas computacionais interconectados. Inicialmente, o foco estava na proteção de sistemas isolados, no entanto, com a popularização da internet nos anos 1990, ataques cibernéticos se intensificaram, levando ao desenvolvimento de ferramentas de detecção, como antivírus e *firewalls* (DAVIES, 2021).

Com a crescente complexidade do ambiente digital e a evolução das ameaças, a cibersegurança passou a enfrentar desafios mais sofisticados. A adoção generalizada da computação em nuvem aumentou a complexidade do gerenciamento de redes e elevou riscos como configurações incorretas, APIs mal protegidas e outros pontos exploráveis

por *hackers*. O crescimento do trabalho remoto e híbrido, aliado a políticas de BYOD (*Bring Your Own Device*), ampliou o número de conexões, dispositivos e dados que precisam ser protegidos. Paralelamente, a proliferação da Internet das Coisas (IoT) trouxe dispositivos frequentemente inseguros ou mal configurados, facilmente sequestráveis por agentes mal-intencionados. Mais recentemente, a ascensão da inteligência artificial, especialmente a generativa, criou um cenário de ameaças inédito, no qual técnicas como a injeção de *prompts* têm sido exploradas por *hackers*, ainda sem proteção adequada em muitas iniciativas (IBM, 2024a).

Para enfrentar esses desafios, a inteligência artificial tem se tornado uma ferramenta essencial na cibersegurança. Suas aplicações estão sendo incorporadas em Plataformas de Inteligência de Ameaças (TIPs) e sistemas de Orquestração, Automação e Resposta de Segurança (SOAR), com o objetivo de automatizar a triagem de alertas e correlacionar Indicadores de Comprometimento (IoCs) em tempo real. No contexto da segurança digital, a IA pode ser utilizada para analisar *logs*, prever possíveis ameaças, identificar vulnerabilidades e até mesmo reconhecer padrões de comportamento que possam indicar atividades anômalas. Técnicas de aprendizado de máquina permitem que os sistemas detectem desvios do comportamento esperado, enquanto a *Cyber Threat Intelligence* (CTI) utiliza a IA para organizar e analisar informações sobre ameaças de forma mais rápida e eficiente. Além disso, a IA ajuda a automatizar tarefas repetitivas, liberando os profissionais de segurança para se concentrarem em decisões estratégicas (RED HAT, 2024). Dessa forma, a IA fortalece a proteção de sistemas e redes, promovendo uma abordagem mais proativa e ágil diante das constantes mudanças e desafios do cenário cibernético.

2.3 Inteligência de Ameaças Cibernéticas

Conforme descrito por Rahman, Mahdavi-Hezaveh e Williams (2021), apesar de não haver uma definição universalmente aceita para a *Cyber Threat Intelligence* (CTI), entende-se como tal qualquer informação útil à segurança digital que permita antecipar, evitar ou reagir a ataques cibernéticos. Ainda nesse contexto, Dalziel (2014) aponta que a CTI refere-se a dados que passaram por algum nível de análise, filtragem ou interpretação e que possuem relevância prática para uma organização, apoiando tomadas de decisão e respostas concretas. Dessa forma, profissionais de cibersegurança podem interpretar os riscos e planejar respostas defensivas adequadas.

Para organizar essa prática, Rahman, Mahdavi-Hezaveh e Williams (2021) classificam a CTI em quatro subdomínios principais. Esses subdomínios se complementam, formando um ciclo integrado de coleta, análise e aplicação de inteligência:

- **CTI Estratégica:** voltada a tomadores de decisão, apresenta informações em re-

latórios e *briefings* de alto nível, incluindo impactos financeiros, tendências e dados históricos sobre ataques.

- **CTI Operacional:** fornece informações sobre ataques iminentes contra uma organização, como a natureza, o momento e os atores envolvidos, sendo aplicada em centros de operações de segurança para respostas imediatas.
- **CTI Tática:** descreve Táticas, Técnicas e Procedimentos (TTPs) de adversários, explicando os objetivos, métodos e ferramentas utilizadas, o que possibilita aprimorar estratégias defensivas.
- **CTI Técnica:** oferece informações em nível granular, como indicadores de comprometimento (IoCs), endereços IP, URLs, *hashes* de *malware* e padrões de tráfego, que podem ser diretamente processados por ferramentas de monitoramento e defesa.

Com base nesse entendimento, a CTI assume um papel essencial na antecipação de riscos e na implementação de ações preventivas mais eficazes no combate às ameaças digitais.

No entanto, diante da constante evolução do cibercrime, os indicadores tradicionais não devem ser considerados isoladamente na formulação de estratégias de defesa, uma vez que possuem natureza volátil e podem rapidamente se tornar obsoletos ou inválidos, como apontam [Iklody et al. \(2018\)](#). Nesse contexto, conforme destaca [CYBLE \(2025\)](#), a adoção de tecnologias emergentes, como inteligência artificial e aprendizado de máquina, viabiliza análises em tempo real e identificação de comportamentos anômalos com maior precisão. Além disso, tais tecnologias oferecem capacidades preditivas, permitindo reconhecer padrões em dados históricos e antecipar possíveis ataques e vulnerabilidades. Outro ponto de destaque é a automação, que possibilita o processamento eficiente de grandes volumes de dados, reduzindo tarefas repetitivas e ampliando a eficácia das equipes de segurança. Integradas à CTI, essas inovações fortalecem a postura defensiva das organizações e contribuem para uma resposta mais ágil e estratégica frente às ameaças emergentes.

2.4 *Surface Web, Deep Web e Dark Web*

A Internet, reconhecida como o maior sistema de informações em escala mundial, é composta por diferentes camadas que se distinguem principalmente pelo nível de acesso e visibilidade dos conteúdos. Essas camadas são conhecidas como *Surface Web*, *Deep Web* e *Dark Web*, e serão descritas a seguir.

A *Surface Web* representa a parte da Internet facilmente acessível e indexada pelos motores de busca tradicionais, como Google e Bing ([BASHEER; ALKHATIB, 2021](#)). A

criação dessa camada está diretamente ligada ao surgimento do primeiro navegador, o WorldWideWeb, desenvolvido por Tim Berners-Lee em 1990, que tornou possível acessar conteúdos de forma visual e organizada (CAMILLERI, 2020). Desde então, a *Surface Web* tem evoluído de páginas estáticas de texto para conteúdos multimídia interativos, especialmente após a adoção do HTML5, que dispensou *plugins* como o Flash, aprimorando a experiência do usuário e ampliando o alcance e a diversidade das informações disponíveis (RANKING ARTICLES, 2023). Paralelamente, técnicas de *Search Engine Optimization* (SEO) passaram a ser empregadas para melhorar a indexação e a visibilidade de páginas, por meio de estratégias como otimização *on-page*, seleção de palavras-chave e adaptação para dispositivos móveis (KATUMBA; COETZEE, 2017). Exemplos típicos da *Surface Web* incluem redes sociais, *blogs*, sites de notícias, *e-commerces*, entre outros.

Apesar de sua ampla visibilidade, a *Surface Web* representa apenas uma pequena fração da totalidade dos dados disponíveis na Internet, correspondendo a cerca de 10% da rede (PROXIMONIVEL, 2025). Por trás dela encontra-se a *Deep Web*, composta por conteúdos não indexados pelos mecanismos tradicionais de busca (BASHEER; ALKHATIB, 2021). O termo *Deep Web* foi introduzido por Bergman (2001), que evidenciou a imensa quantidade de informações que permanecem invisíveis aos motores de busca convencionais, estimada em 400 a 550 vezes o tamanho da *Surface Web*.

Desde então, a *Deep Web* tem passado por avanços importantes, especialmente no desenvolvimento de técnicas para facilitar a navegação e recuperação de dados em bancos de dados dinâmicos e não indexados. Uma dessas inovações envolve o uso de métodos de Aprendizado por Reforço para tornar os *crawlers* mais eficientes na exploração desse conteúdo oculto (JIANG et al., 2010).

Na prática, a *Deep Web* engloba conteúdos usualmente protegidos por autenticação, como contas bancárias *online*, e-mails privados, bases de dados acadêmicas por assinatura, comunicados médicos ou legais reservados, além de redes internas de empresas ou órgãos governamentais, todos inacessíveis por meio de mecanismos de busca convencionais (GROUP-IB, 2025).

Dentro da *Deep Web*, existe uma camada ainda mais específica e restrita, conhecida como *Dark Web*. Esta representa uma porção da Internet propositalmente oculta, acessível somente por meio de *softwares* especializados que garantem anonimato e dificultam a identificação dos usuários, como o *The Onion Router* (TOR) (BASHEER; ALKHATIB, 2021). Conforme explica Bartlett (2015), a tecnologia que deu origem à *Dark Web* teve início como um projeto do Laboratório de Pesquisa Naval dos Estados Unidos, com o objetivo de permitir comunicações anônimas e seguras entre agentes governamentais. Com o passar do tempo, essa tecnologia evoluiu para a rede Tor, atualmente utilizada mundialmente para navegação anônima, proteção da identidade e privacidade dos usuários. No entanto, essa garantia de anonimato acarreta a predominância de atividades

maliciosas e criminosas nesse ambiente. Exemplos notórios dessas atividades, como destacam [Basheer e Alkhatib \(2021\)](#), incluem tráfico de drogas e armas, tráfico de pessoas, pornografia infantil e fraudes.

3 Trabalhos Relacionados

Este capítulo dedica-se à revisão dos trabalhos que forneceram os alicerces teóricos e práticos para a concepção deste projeto. Como o ThreatLens foi desenvolvido para ser a camada de apresentação visual de um ecossistema mais amplo de inteligência de ameaças cibernéticas, a análise a seguir divide-se em duas frentes complementares. A primeira busca contextualizar a origem, o processamento e o domínio das informações consumidas pelo *dashboard*, analisando três sistemas desenvolvidos no âmbito deste macroprojeto: o modelo de classificação estruturado por Jesus Filho (2023), o modelo de agrupamento temático de Reis (2024) e a infraestrutura de extração de dados de Vieira (2025). A segunda frente, por sua vez, dedica-se a propostas da literatura externa voltadas especificamente para a visualização de dados de segurança, abordando as soluções gráficas e interativas desenvolvidas por Böhm, Menges e Pernul (2018) e Lughbi, Mars e Almotairi (2024).

O trabalho desenvolvido por Jesus Filho (2023) teve como objetivo principal desenvolver um modelo de classificação capaz de identificar automaticamente publicações maliciosas em fóruns da *dark web*, contribuindo para a detecção precoce de ameaças cibernéticas. Para isso, foram aplicadas técnicas de Processamento de Linguagem Natural (PLN) e aprendizado de máquina, realizando a coleta de dados de fóruns e a extração de características textuais para treinar classificadores supervisionados. Os resultados mostraram que o modelo proposto obteve boa performance na distinção entre *posts* legítimos e maliciosos, com métricas relevantes de acurácia e precisão. O trabalho de Jesus Filho (2023) se relaciona com este porque, enquanto ele foca na identificação automática das ameaças, este projeto busca aprimorar a análise e visualização desses dados por meio de um *dashboard* interativo. Dessa forma, ambos contribuem para a área de segurança cibernética, sendo complementares: a coleta de dados realizada por Jesus Filho (2023) fornece a base de dados e este trabalho atua na exploração visual e analítica dessas informações.

Por outro lado, o trabalho de Reis (2024) teve como objetivo analisar *posts* maliciosos extraídos de fóruns da *Dark Web*, utilizando técnicas de aprendizado de máquina não supervisionado para identificar padrões temáticos relacionados a ameaças cibernéticas. A metodologia empregou algoritmos de agrupamento, como *K-means*, DBSCAN e KNN, além da modelagem de tópicos *Latent Dirichlet Allocation* (LDA), aplicados a uma base de mais de 11 mil *posts*, previamente processados e vetorizados com TF-IDF. Os resultados demonstraram que o *K-means* obteve melhor desempenho, estruturando os dados em três *clusters* principais que revelaram temas como segurança de dados e contas, busca por informações pessoais e sensíveis, e a formação de comunidades de *hacking*. O estudo evidenciou como técnicas de análise automática de texto podem auxiliar na compreensão das táticas e intenções de cibercriminosos, contribuindo para a área de *Cyber Threat In-*

telligence (CTI). Este trabalho se relaciona diretamente ao de Reis (2024), uma vez que, apesar do foco deste trabalho não ser a extração dos dados, ambos buscam organizar e apresentar informações relevantes sobre ameaças cibernéticas.

Por fim, o trabalho de Vieira (2025) teve como objetivo desenvolver um *bot* para captura e análise de mensagens provenientes de canais públicos do Telegram, visando apoiar a inteligência de ameaças cibernéticas. O sistema foi projetado para coletar mensagens de forma automática e manual, armazenando-as em uma base estruturada e possibilitando sua posterior classificação quanto à relevância, utilizando aprendizado de máquina supervisionado. Nos experimentos, foram coletadas 760 mensagens, das quais cerca de 16% foram identificadas como relevantes para o contexto de cibersegurança. O trabalho de Vieira (2025) relaciona-se a este projeto na medida em que sua infraestrutura atua como a fonte de coleta no Telegram, fornecendo os dados que são posteriormente processados e transformados em informações visuais e gerenciais pelo *dashboard*.

A análise dos trabalhos desenvolvidos no âmbito deste macroprojeto evidencia uma concentração nas etapas de coleta, classificação e organização estrutural das informações. Para complementar esse ecossistema, faz-se necessário analisar propostas na literatura externa que, assim como o ThreatLens, atuam na camada de visualização de dados de segurança.

Nesse contexto visual, Böhm, Menges e Pernul (2018) propuseram uma abordagem de *visual analytics* para *Cyber Threat Intelligence* (CTI), visando reduzir a complexidade na interpretação humana de dados de segurança. A solução apresenta uma interface interativa baseada em grafos para explorar as relações entre entidades de CTI. Embora ambas as soluções utilizem a visualização para facilitar a análise, o trabalho de Böhm, Menges e Pernul (2018) foca especificamente em mapeamento relacional via grafos, enquanto o ThreatLens adota uma abordagem de múltiplos componentes (indicadores, gráficos, tabelas e filtros) para representar dados de diversas origens.

De maneira ainda mais alinhada à proposta deste projeto, Lughbi, Mars e Almotairi (2024) desenvolveram um *dashboard* interativo que combina PLN e recursos visuais (mapas geográficos, tabelas e gráficos) para classificar publicações sobre ataques cibernéticos provenientes da plataforma X (antigo Twitter). A principal distinção entre os sistemas reside na abrangência arquitetural dos dados: enquanto a solução dos autores é restrita a uma única rede social, o ThreatLens foi concebido como uma plataforma integradora capaz de processar, filtrar e analisar informações provenientes de múltiplas fontes e diferentes etapas de detecção do ecossistema de pesquisa.

Em suma, enquanto os trabalhos de Jesus Filho (2023), Reis (2024) e Vieira (2025) viabilizam a obtenção e o enriquecimento dos dados de base, as abordagens de Böhm, Menges e Pernul (2018) e Lughbi, Mars e Almotairi (2024) validam a importância acadêmica de interfaces gráficas para inteligência de ameaças. O ThreatLens posiciona-se na

intersecção dessas necessidades, consolidando as informações complexas do ecossistema local e entregando-as de forma visual, interativa e organizada ao usuário final.

4 ThreatLens: plataforma para análise de dados

O ThreatLens é uma plataforma web desenvolvida no contexto de um projeto de pesquisa em Inteligência de Ameaças Cibernéticas (*Cyber Threat Intelligence* – CTI). Seu propósito é disponibilizar, de maneira organizada e acessível, informações sobre possíveis ameaças identificadas em fontes abertas da Internet. A plataforma integra a etapa final de um processo mais amplo de obtenção, processamento e análise de dados, permitindo que informações de diferentes origens sejam consultadas por meio de uma interface unificada.

A concepção da ferramenta está ancorada em trabalhos anteriores da Faculdade de Computação da Universidade Federal de Uberlândia (FACOM/UFU), que desenvolveram mecanismos para a coleta de dados, classificação de publicações e identificação de padrões. O ThreatLens atua como a camada de apresentação desse ecossistema, reunindo os resultados produzidos por essas etapas de *back-end* em um ambiente voltado à análise visual e exploratória.

4.1 Visão geral da plataforma

O sistema reúne informações provenientes de diferentes etapas do processamento dos dados, permitindo que os resultados sejam consultados sem a necessidade de acesso direto às bases de dados. As publicações disponibilizadas pela plataforma possuem informações relacionadas à sua origem, conteúdo e classificação, possibilitando a consulta tanto do conjunto de dados quanto de registros específicos.

A plataforma também apresenta informações agregadas sobre as publicações, como estatísticas e representações gráficas. Esses recursos permitem observar características gerais dos dados, identificar padrões e selecionar informações de interesse para uma análise mais detalhada. Dessa forma, o usuário pode partir de uma visão geral dos dados e, quando necessário, consultar individualmente as publicações relacionadas.

O acesso às informações ocorre a partir dos dados processados pelas etapas anteriores do projeto. As publicações são coletadas em fontes abertas e posteriormente submetidas a processos de classificação e análise, cujos resultados são utilizados pelo ThreatLens. Assim, a plataforma não realiza a coleta ou o processamento original dos dados, mas disponibiliza os resultados dessas etapas para consulta e análise.

4.2 Componentes do sistema

O ThreatLens é composto por dois componentes principais: o *back-end* e o *front-end*. O *back-end* é responsável por disponibilizar os dados e os recursos utilizados pela plataforma, incluindo autenticação, controle de acesso e consulta às informações processadas. Já o *front-end* corresponde ao *dashboard* utilizado pelos usuários para consultar e visualizar esses dados de forma interativa.

A plataforma utiliza dois bancos de dados PostgreSQL. O banco **asgard** armazena os dados coletados e processados pelo pipeline e é utilizado pelo ThreatLens para realizar consultas. O segundo banco é utilizado pela própria aplicação para armazenar informações relacionadas aos usuários, como credenciais, tokens de sessão e códigos de verificação de e-mail. Essa separação mantém os dados de inteligência independentes das informações de gerenciamento da aplicação.

4.3 Público-alvo e finalidade

O sistema é direcionado a profissionais e pesquisadores envolvidos com segurança da informação e CTI que necessitam acompanhar e analisar informações relacionadas ao cenário de ameaças digitais. Nesse contexto, a plataforma foi desenvolvida considerando a necessidade de consultar diferentes registros e obter uma visão geral dos dados processados pelo projeto.

A combinação entre informações consolidadas e registros individuais permite diferentes níveis de análise. Os indicadores e gráficos auxiliam na compreensão do conjunto de dados, enquanto a consulta às publicações possibilita examinar informações específicas. Dessa forma, o ThreatLens oferece uma interface única para exploração dos dados produzidos pelo projeto, reduzindo a dependência de consultas diretas às estruturas de armazenamento.

5 Tecnologias Utilizadas

Neste capítulo, são apresentadas as tecnologias utilizadas no desenvolvimento deste trabalho. Para proporcionar uma compreensão estruturada do ecossistema do projeto, a Seção 5.1 introduz o Figma, ferramenta utilizada na prototipação da interface. Em seguida, as Seções 5.2 e 5.3 detalham o uso da biblioteca React e do React Router DOM na construção e roteamento da aplicação. A estruturação visual e a renderização de dados são abordadas nas Seções 5.4 e 5.5, que descrevem o emprego do CSS e do Chart.js, respectivamente. Por fim, as Seções 5.6 a 5.8 apresentam as ferramentas de infraestrutura, desenvolvimento e controle de versão que deram suporte ao projeto, englobando Git, GitHub, VS Code e Vite.

5.1 Figma

O Figma é uma ferramenta de design de interfaces baseada em nuvem, desenvolvida pela Figma, Inc., que permite a criação de wireframes, protótipos e layouts de alta fidelidade diretamente no navegador (FIGMA, 2026). Por ser uma aplicação web, dispensa a instalação de software local e possibilita o acesso ao projeto de qualquer dispositivo, além de oferecer recursos de colaboração em tempo real entre membros de uma equipe.

No desenvolvimento do ThreatLens, o Figma foi utilizado na etapa de prototipação da interface, permitindo definir a estrutura visual das telas, a paleta de cores, a tipografia e a disposição dos componentes antes da implementação. Essa etapa antecedeu a codificação e serviu como referência durante o desenvolvimento, reduzindo retrabalho e tornando as decisões de design mais deliberadas. A ferramenta foi utilizada em sua versão gratuita, disponível em plataforma web.

5.2 React

React é uma biblioteca JavaScript de código aberto desenvolvida pelo Facebook voltada para a construção de interfaces de usuário (META, 2024). Seu principal conceito é a componentização: a interface é dividida em componentes independentes e reutilizáveis, cada um responsável por uma parte da tela, o que facilita a organização, manutenção e reaproveitamento de código ao longo do projeto.

A biblioteca utiliza um mecanismo denominado Virtual DOM, que consiste em uma representação em memória da árvore de elementos da interface. Sempre que o estado de

um componente é alterado, o React calcula a diferença entre o Virtual DOM atual e o anterior e atualiza no DOM real apenas os elementos que sofreram mudanças, tornando as atualizações de tela eficientes (REACT, 2024). Além disso, o modelo baseado em estado e propriedades (props) permite que dados fluam de forma previsível entre os componentes da aplicação.

O React foi escolhido para este trabalho por sua produtividade na criação de Single Page Applications (SPAs), seu ecossistema de bibliotecas complementares e sua ampla adoção na indústria, o que garante vasta documentação e suporte da comunidade. A versão utilizada foi a 18.3.1.

5.3 React Router DOM

O React Router DOM é uma biblioteca de código aberto para roteamento no lado do cliente em aplicações React (REACT ROUTER, 2024). Em uma Single Page Application, não há recarregamento de página a cada navegação. Nesse modelo, o roteamento é gerenciado pelo próprio JavaScript no navegador, que atualiza a URL e renderiza o componente correspondente sem solicitar um novo documento ao servidor. O React Router DOM destaca-se como uma das soluções mais utilizadas para essa finalidade no ecossistema React, fornecendo componentes declarativos que associam caminhos de URL a componentes da interface.

A biblioteca oferece recursos como rotas aninhadas, parâmetros dinâmicos de URL, navegação programática e proteção de rotas, permitindo estruturar a aplicação em múltiplas páginas com transições fluidas para o usuário. No ThreatLens, o React Router DOM foi utilizado para gerenciar a navegação entre as diferentes seções da aplicação, como cadastro, login, painel principal e publicações. A versão utilizada foi a 6.26.2.

5.4 CSS

CSS é a linguagem de folha de estilos utilizada para descrever a apresentação visual de documentos HTML (W3C, 2021). Por meio de seletores e propriedades, é possível definir cores, tipografia, espaçamento, posicionamento e animações de todos os elementos da interface. O mecanismo de cascata determina a prioridade entre regras conflitantes com base em especificidade e ordem de declaração.

Neste trabalho, o CSS foi utilizado diretamente, sem a intermediação de frameworks de interface como Bootstrap ou Tailwind, o que conferiu controle total sobre o design visual da aplicação. Foram empregadas variáveis CSS (custom properties) para centralizar as definições de cores e tamanhos do sistema de design, e media queries para adaptar a interface a diferentes resoluções de tela, garantindo responsividade. A organi-

zação dos estilos foi realizada por meio de arquivos CSS modulares, associados individualmente a cada componente React.

5.5 Chart.js

Chart.js é uma biblioteca JavaScript de código aberto para a criação de gráficos interativos renderizados sobre o elemento canvas do HTML5 ([CHART.JS, 2026](#)). A biblioteca oferece suporte a diversos tipos de gráficos, como barras, linhas, pizza, rosca, radar e dispersão, todos com animações e interatividade nativas, como exibição de tooltips ao passar o cursor sobre os dados.

A integração do Chart.js com o React foi realizada por meio da biblioteca `react-chartjs-2`, que encapsula os gráficos como componentes React, simplificando o gerenciamento do ciclo de vida dos gráficos dentro do modelo de componentes da aplicação. O Chart.js foi escolhido neste trabalho pela facilidade de uso, pela variedade de tipos de gráfico disponíveis, pela responsividade nativa e pela qualidade de sua documentação. A versão utilizada foi a 4.4.4.

5.6 Git e GitHub

Git é um sistema de controle de versão distribuído de código aberto, criado por Linus Torvalds em 2005 ([CHACON; STRAUB, 2014](#)). Ele registra o histórico de todas as alterações realizadas nos arquivos de um projeto, permitindo que o desenvolvedor visualize o que foi modificado, quando e por quem, além de possibilitar a reversão para qualquer versão anterior do código. O Git opera com o conceito de *branches*, que são ramificações independentes do histórico de desenvolvimento. Isso permite trabalhar em novas funcionalidades ou correções de forma isolada, sem interferir no código principal, e posteriormente integrar as mudanças por meio de operações de *merge*.

Para hospedar e gerenciar esse versionamento, utilizou-se o GitHub, uma plataforma web de hospedagem de repositórios Git que adiciona funcionalidades de colaboração e gerenciamento de projetos ([GITHUB, 2026](#)). A plataforma oferece recursos como *pull requests*, que permitem revisar e discutir mudanças antes de integrá-las ao código principal, e *issues*, utilizadas para registrar tarefas e problemas.

Neste trabalho, o Git foi utilizado localmente para rastrear as mudanças realizadas durante o desenvolvimento da aplicação, mantendo um histórico estruturado. Em conjunto, o GitHub operou como repositório remoto do projeto, servindo como mecanismo de backup do código-fonte e possibilitando o acesso seguro à aplicação a partir de diferentes ambientes de desenvolvimento.

5.7 VS Code

Visual Studio Code é um editor de código-fonte gratuito, leve e extensível desenvolvido pela Microsoft, disponível para os sistemas operacionais Windows, macOS e Linux ([MICROSOFT, 2026](#)). Apesar de ser um editor e não uma IDE completa, oferece recursos como realce de sintaxe, conclusão de código inteligente, depuração integrada e terminal embutido.

Uma das principais características do Visual Studio Code é o seu amplo e ativo ecossistema de extensões, que permite adicionar suporte a linguagens, linters, formata-dores e integrações diversas. Para o desenvolvimento deste trabalho, foram utilizadas extensões como o suporte ao ESLint, para análise estática do código JavaScript, e inte-grações com o Git diretamente pela interface do editor. O Visual Studio Code foi escolhido por ser um dos editores mais adotados pela comunidade de desenvolvimento JavaScript e React, contando com documentação abundante e extensões específicas para o ecossistema utilizado neste projeto.

5.8 Vite

Vite é uma ferramenta de construção e servidor de desenvolvimento para projetos *front-end*, criada por Evan You ([VITE, 2026](#)). Diferentemente de ferramentas mais antigas como o Webpack, o Vite não realiza o empacotamento (bundling) completo do código durante o desenvolvimento. Em vez disso, aproveita o suporte nativo dos navegadores modernos a ES Modules para servir os arquivos diretamente, resultando em um servidor de desenvolvimento com rápida inicialização independentemente do tamanho do projeto.

O Vite também oferece o recurso de Hot Module Replacement (HMR), que subs-titui no navegador apenas o módulo alterado pelo desenvolvedor sem recarregar a página inteira, preservando o estado da aplicação durante o desenvolvimento. Para a geração do pacote de produção, a ferramenta realiza o empacotamento de forma automática, gerando artefatos finais otimizados e com tamanho reduzido. O Vite foi adotado neste trabalho por proporcionar um ciclo de desenvolvimento ágil, além de oferecer configuração simplificada para projetos React. A versão utilizada foi a 6.3.5.

6 Desenvolvimento

Este capítulo aborda o processo de planejamento e criação do *front-end* da plataforma, detalhando as etapas executadas desde a concepção até a conclusão do sistema. Para apresentar o fluxo de trabalho de maneira estruturada, a Seção 6.1 descreve o levantamento de requisitos funcionais e não funcionais que guiaram a construção da aplicação. Em seguida, a Seção 6.2 apresenta a etapa de prototipação de alta fidelidade, responsável por planejar e validar a identidade visual e a disposição dos elementos da interface. Por fim, a Seção 6.3 detalha a fase de implementação prática, englobando desde a criação do projeto e a estruturação de diretórios até o desenvolvimento dos componentes, páginas e a integração final com o *back-end*.

Todo o código-fonte desenvolvido neste trabalho encontra-se disponível no repositório GitHub, podendo ser acessado por meio do endereço <<https://github.com/pmiguell/threatlens>>. O repositório contém a implementação completa da aplicação desenvolvida e seu histórico de versionamento.

6.1 Levantamento de requisitos

Antes de iniciar a implementação do *front-end* da plataforma, foi realizada a definição dos requisitos do sistema, etapa responsável por identificar as principais funcionalidades e características esperadas para a aplicação. Para garantir que o *dashboard* atendesse às necessidades do macroprojeto de inteligência de ameaças no qual está inserido, o processo de elicitação de requisitos baseou-se em uma abordagem híbrida, combinando análise documental e reuniões de alinhamento.

A análise documental consistiu no estudo dos dados produzidos pelas etapas de coleta e classificação das pesquisas integradas ao ecossistema, permitindo mapear exatamente quais informações precisavam ser consumidas e traduzidas visualmente. Em complemento, foram conduzidas reuniões periódicas com o orientador e o grupo de pesquisa para debater e refinar as necessidades dos usuários finais da plataforma, priorizando a simplicidade e o desenvolvimento apenas das funcionalidades estritamente necessárias ao escopo visual.

Essa definição teve como objetivo orientar a construção das telas, dos fluxos de navegação e dos recursos presentes na plataforma, considerando as necessidades propostas para o projeto. Os requisitos levantados foram organizados em Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF).

6.1.1 Requisitos Funcionais

Os requisitos funcionais estabelecem as operações e as capacidades técnicas suportadas pelo sistema, definindo as funcionalidades necessárias para atender aos objetivos da plataforma. Os quadros a seguir apresentam os requisitos funcionais identificados para o desenvolvimento do ThreatLens.

Quadro 1 – RF01: Cadastro de usuário

Requisito	RF01
Nome	Cadastro de usuário
Descrição	A plataforma deve permitir que o usuário crie uma conta fornecendo suas informações para acesso.

Fonte: Autoria própria.

Quadro 2 – RF02: Autenticação

Requisito	RF02
Nome	Autenticação
Descrição	A plataforma deve permitir que o usuário realize autenticação através de e-mail e senha.

Fonte: Autoria própria.

Quadro 3 – RF03: Redefinição de senha

Requisito	RF03
Nome	Redefinição de senha
Descrição	A plataforma deve permitir que o usuário redefina sua senha através do e-mail fornecido no momento do cadastro.

Fonte: Autoria própria.

Quadro 4 – RF04: Logout

Requisito	RF04
Nome	Logout
Descrição	A plataforma deve permitir que o usuário encerre sua sessão atual, exigindo autenticação novamente para acesso.

Fonte: Autoria própria.

Quadro 5 – RF05: Consulta de perfil e alteração de senha

Requisito	RF05
Nome	Consulta de perfil e alteração de senha
Descrição	A plataforma deve permitir que o usuário autenticado consulte seu nome de usuário e e-mail e altere sua senha. O nome de usuário e o e-mail devem ser exibidos apenas para consulta, sem possibilidade de edição nessa tela.

Fonte: Autoria própria.

Quadro 6 – RF06: Carregamento de publicações

Requisito	RF06
Nome	Carregamento de publicações
Descrição	A plataforma deve carregar as informações dos posts coletados pelo sistema de detecção de ameaças cibernéticas.

Fonte: Autoria própria.

Quadro 7 – RF07: Busca de publicações

Requisito	RF07
Nome	Busca de publicações
Descrição	A plataforma deve permitir que o usuário realize buscas de posts através do título.

Fonte: Autoria própria.

Quadro 8 – RF08: Ordenação de publicações

Requisito	RF08
Nome	Ordenação de publicações
Descrição	A plataforma deve permitir que o usuário realize ordenação de posts através de seus atributos, como identificador, fonte, classificação e conteúdo.

Fonte: Autoria própria.

Quadro 9 – RF09: Visualização de *dashboard*

Requisito	RF09
Nome	Visualização de <i>dashboard</i>
Descrição	A plataforma deve apresentar um painel contendo indicadores, estatísticas e informações relacionadas às ameaças cibernéticas identificadas.

Fonte: Autoria própria.

Quadro 10 – RF10: Navegação entre páginas

Requisito	RF10
Nome	Navegação entre páginas
Descrição	A plataforma deve permitir a navegação entre suas diferentes seções sem recarregamento completo da página.

Fonte: Autoria própria.

Quadro 11 – RF11: Exibição de mensagens de feedback

Requisito	RF11
Nome	Exibição de mensagens de feedback
Descrição	A plataforma deve informar ao usuário o resultado das operações realizadas, como sucesso ou falha em ações executadas.

Fonte: Autoria própria.

Quadro 12 – RF12: Remoção de contas

Requisito	RF12
Nome	Remoção de contas
Descrição	A plataforma deve permitir que o administrador delete contas criadas na plataforma.

Fonte: Autoria própria.

Quadro 13 – RF13: Alternância de tema

Requisito	RF13
Nome	Alternância de tema
Descrição	A plataforma deve permitir que o usuário alterne a interface visual entre os modos claro e escuro.

Fonte: Autoria própria.

6.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem critérios relacionados à qualidade e ao comportamento da aplicação, estabelecendo características que influenciam a experiência do usuário e o funcionamento do sistema. Os quadros a seguir apresentam os requisitos não funcionais definidos para o desenvolvimento do ThreatLens.

Quadro 14 – RNF01: Responsividade

Requisito	RNF01
Nome	Responsividade
Descrição	A interface da plataforma deve adaptar-se adequadamente a diferentes tamanhos de tela, incluindo desktops, tablets e dispositivos móveis.

Fonte: Autoria própria.

Quadro 15 – RNF02: Segurança

Requisito	RNF02
Nome	Segurança
Descrição	A plataforma deve garantir que apenas usuários autenticados possam acessar áreas restritas do sistema.

Fonte: Autoria própria.

Quadro 16 – RNF03: Usabilidade

Requisito	RNF03
Nome	Usabilidade
Descrição	A plataforma deve possuir interface intuitiva e organizada, facilitando a navegação e utilização por parte dos usuários.

Fonte: Autoria própria.

Quadro 17 – RNF04: Manutenibilidade

Requisito	RNF04
Nome	Manutenibilidade
Descrição	A aplicação deve possuir organização padronizada do código, permitindo facilidade na manutenção e expansão de funcionalidades futuras.

Fonte: Autoria própria.

6.2 Prototipação de alta fidelidade

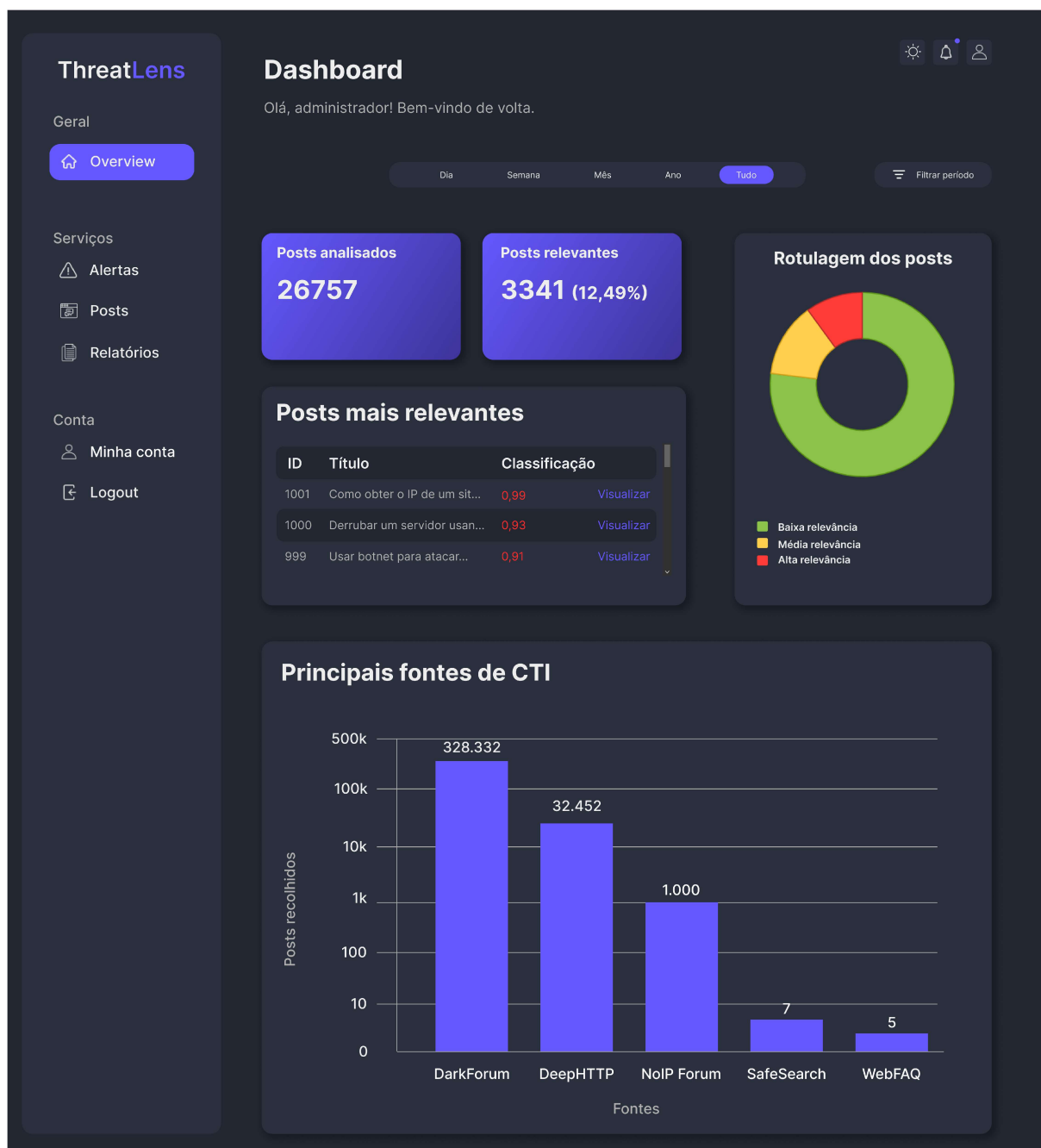
A etapa de prototipação de alta fidelidade teve como objetivo definir visualmente a interface da plataforma antes do início da implementação do *front-end*, de forma a reduzir retrabalho durante essa etapa. Diferentemente de protótipos de baixa fidelidade, que representam apenas estruturas básicas e fluxos de navegação, os protótipos de alta fidelidade simulam de maneira mais próxima a aparência e o comportamento esperados da aplicação final.

Para o desenvolvimento dos protótipos foi utilizada a ferramenta Figma, permitindo a criação das principais telas do sistema, como login, cadastro, painel principal e publicações. Nessa etapa, foram definidos elementos visuais como cores, tipografia, espaçamento e organização dos componentes. Além disso, o mapeamento inicial incluiu o planejamento da responsividade para telas menores, garantindo que o comportamento adaptativo da interface fosse validado antes da codificação.

Como o código do *front-end* foi implementado com alta fidelidade ao que foi especificado no sistema de design da ferramenta, optou-se por apresentar uma tela exemplar para ilustrar o resultado dessa etapa de planejamento estético. A Figura 1 apresenta

o protótipo de alta fidelidade desenvolvido para o *dashboard* da plataforma. Os dados apresentados no protótipo são meramente ilustrativos e foram utilizados apenas para representar visualmente o funcionamento e a disposição das informações na interface, não correspondendo aos dados reais presentes na base do sistema.

Figura 1 – Protótipo de alta fidelidade do *dashboard* desenvolvido no Figma



Fonte: Autoria própria.

6.3 Etapas da implementação

Com a conclusão do levantamento de requisitos e a validação visual por meio dos protótipos de alta fidelidade, deu-se início à fase de desenvolvimento de software da plataforma ThreatLens. Esta etapa compreendeu a tradução das especificações técnicas e funcionais em código-fonte, utilizando o ecossistema de ferramentas e tecnologias previamente fundamentadas.

6.3.1 Criação do projeto

A etapa de implementação teve início com a criação do ambiente base da aplicação utilizando o Vite. Por meio de sua interface de linha de comando (CLI), foi gerado o esqueleto inicial do projeto pré-configurado para o ecossistema React.

Após a geração dos arquivos básicos, realizou-se uma limpeza no repositório, removendo estilos e componentes de exemplo nativos da ferramenta. Em seguida, foram instaladas as dependências essenciais para o funcionamento da plataforma, com destaque para o React Router DOM para o gerenciamento de rotas e o Chart.js para a posterior renderização das métricas do *dashboard*.

Para garantir a padronização e a qualidade do código-fonte desde as primeiras etapas, o ESLint foi configurado para atuar na análise estática do JavaScript, identificando precocemente possíveis anomalias sintáticas. Por fim, o repositório local foi inicializado via Git e vinculado a um repositório remoto hospedado no GitHub, estabelecendo o controle de versão que acompanharia todo o ciclo de vida do desenvolvimento do projeto.

6.3.2 Estrutura de diretórios

A estrutura de diretórios foi definida seguindo uma separação baseada em responsabilidades, buscando reduzir o acoplamento entre as diferentes partes da aplicação e facilitar futuras alterações no sistema. O projeto teve início com a seguinte estrutura:

- **/components** - responsável por armazenar componentes reutilizáveis da interface. Essa separação foi adotada para evitar duplicação de código e permitir que elementos visuais comuns fossem utilizados por diferentes páginas da aplicação.
- **/pages** - contém os componentes que representam as páginas completas do sistema, reunindo os elementos necessários para cada tela, como login, *dashboard* e publicações.
- **/services** - destinado aos arquivos responsáveis pela comunicação com serviços externos. Essa camada foi criada para impedir que componentes da interface rea-

lizassem chamadas diretamente à API, mantendo a separação entre apresentação e comunicação com o servidor.

- **/hooks** - armazena hooks personalizados utilizados para encapsular lógicas reutilizáveis da aplicação, como gerenciamento de estados, controle de efeitos e comportamentos compartilhados entre componentes. Esses hooks atuam como uma camada intermediária entre a interface e os serviços definidos em **/services**, realizando o consumo dos dados e disponibilizando-os de forma organizada para os componentes da aplicação.
- **/utils** - contém funções auxiliares utilizadas em diferentes partes da aplicação, como validações, formatações e manipulação de dados.
- **/assets** - responsável por armazenar arquivos estáticos utilizados pela interface, como imagens, ícones e outros recursos visuais.

Essa estrutura permitiu uma divisão organizada das responsabilidades do projeto, separando componentes, páginas, serviços e recursos auxiliares de forma clara. Com essa abordagem, o desenvolvimento se tornou mais modular, facilitando a manutenção, a reutilização de código e a evolução da aplicação ao longo do tempo.

6.3.3 Desenvolvimento de componentes

O desenvolvimento dos componentes da plataforma seguiu uma abordagem baseada em componentização, na qual cada elemento da interface foi implementado como uma unidade independente. Essa decisão foi adotada para favorecer a reutilização de código, reduzir duplicações e facilitar a manutenção da aplicação.

Devido à quantidade de componentes implementados na aplicação, foram selecionados para apresentação nesta seção aqueles considerados mais relevantes para a composição da interface e para o funcionamento das principais funcionalidades da plataforma. Esses componentes podem ser classificados conforme suas funcionalidades dentro da aplicação:

- **Componentes de estrutura e navegação:** Responsáveis pela organização do layout principal da aplicação e pela navegação entre suas seções. Foram desenvolvidos componentes como **Sidebar**, **Navigation** e **Header**, que compõem a estrutura visual das páginas, exibem os menus disponíveis e fornecem acesso às funcionalidades do sistema.
- **Componentes de visualização de dados:** Destinados à apresentação gráfica das informações processadas pela aplicação. Entre eles destacam-se **BarChart**,

DoughnutChart, WordCloud e ProportionCard, utilizados para exibir estatísticas, indicadores e análises relacionadas às publicações coletadas.

- **Componentes de exibição tabular:** Responsáveis pela organização dos dados em formato de tabela, permitindo a consulta das publicações disponíveis. Os componentes `AllPostsTable` e `RelevantPostsTable` apresentam, respectivamente, todas as publicações coletadas e aquelas classificadas como relevantes.
- **Componentes de modais e diálogos:** Utilizados para exibir informações adicionais e realizar ações específicas sem alterar a página atual. Os componentes `PostPopUp` e `ConfirmModal` são responsáveis, respectivamente, pela visualização detalhada de publicações e confirmação de operações.

6.3.4 Desenvolvimento de páginas

As páginas da plataforma ThreatLens foram implementadas a partir da combinação dos componentes desenvolvidos previamente, seguindo a estrutura definida durante a organização do projeto. Cada página é responsável por agrupar os elementos necessários para uma funcionalidade específica da aplicação, enquanto componentes compartilhados, como menu lateral, cabeçalho, tabelas, gráficos, cartões de informação e modais, são reutilizados entre diferentes telas.

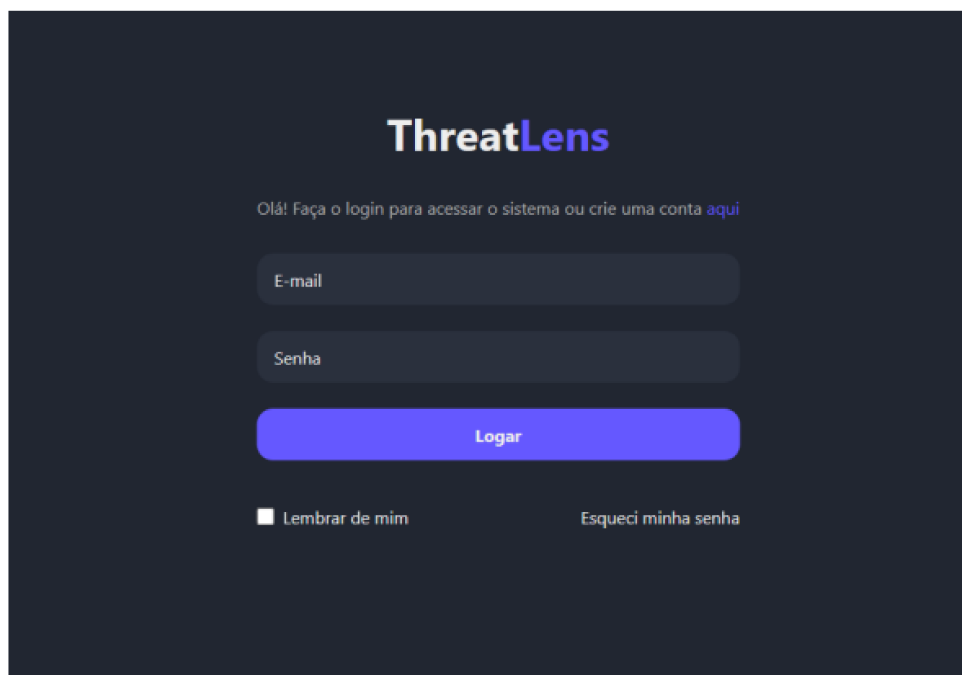
Essa abordagem permitiu manter um padrão visual consistente em toda a aplicação, além de facilitar a manutenção do código, uma vez que alterações realizadas em componentes reutilizados são refletidas automaticamente nas páginas que os utilizam. A separação entre componentes e páginas também contribuiu para manter cada parte do sistema com uma responsabilidade bem definida.

O controle de navegação entre as páginas foi realizado utilizando React Router DOM, adotando uma arquitetura baseada em Single Page Application (SPA). Essa decisão permitiu realizar a troca de telas sem recarregamento completo da aplicação, proporcionando uma navegação mais fluida para o usuário. As rotas foram organizadas em páginas públicas e privadas, garantindo que funcionalidades relacionadas à autenticação, como login, cadastro, confirmação de código e recuperação de senha, pudessem ser acessadas sem uma sessão ativa.

Já as funcionalidades internas da plataforma exigem que o usuário esteja autenticado. Além disso, a área administrativa possui uma restrição adicional, permitindo acesso apenas para usuários com permissão de administrador.

A tela de autenticação apresentada na Figura 2 representa o ponto inicial de acesso ao sistema. Seu desenvolvimento priorizou uma interface simples, contendo apenas os elementos necessários para autenticação e acesso às funcionalidades relacionadas ao cadastro e recuperação de senha.

Figura 2 – Tela de login do ThreatLens

A imagem mostra a interface de login do ThreatLens. No topo, o logo "ThreatLens" é exibido em branco e azul. Abaixo dele, uma mensagem de boas-vindas diz: "Olá! Faça o login para acessar o sistema ou crie uma conta [aqui](#)". Seguem-se dois campos de entrada: "E-mail" e "Senha", ambos com ícones de olho para alternar a visibilidade. Abaixo dos campos, há um botão azul com o texto "Logar". Na base da tela, há uma opção "Lembrar de mim" com uma caixa de seleção e um link "Esqueci minha senha".

ThreatLens

Olá! Faça o login para acessar o sistema ou crie uma conta [aqui](#)

E-mail

Senha

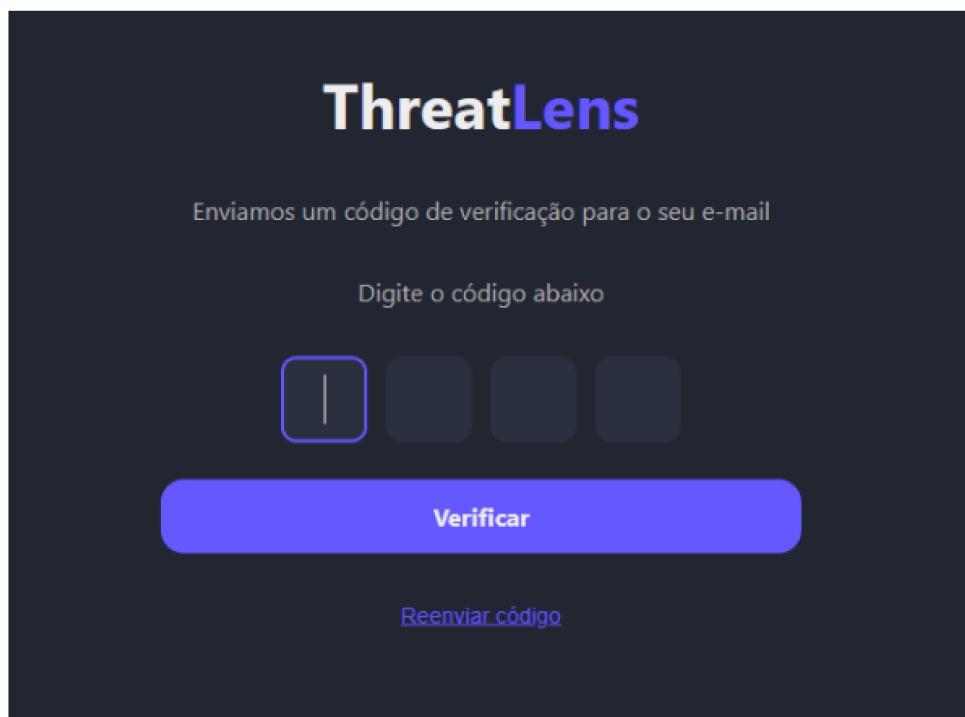
Logar

☐ Lembrar de mim [Esqueci minha senha](#)

Fonte: Autoria própria.

O processo de criação de conta foi implementado utilizando um fluxo dividido em etapas para separar o cadastro inicial da validação de identidade, conforme apresentado na Figura 3. Essa abordagem permitiu reutilizar a mesma lógica de confirmação por código em funcionalidades como recuperação de senha.

Figura 3 – Tela de verificação de código

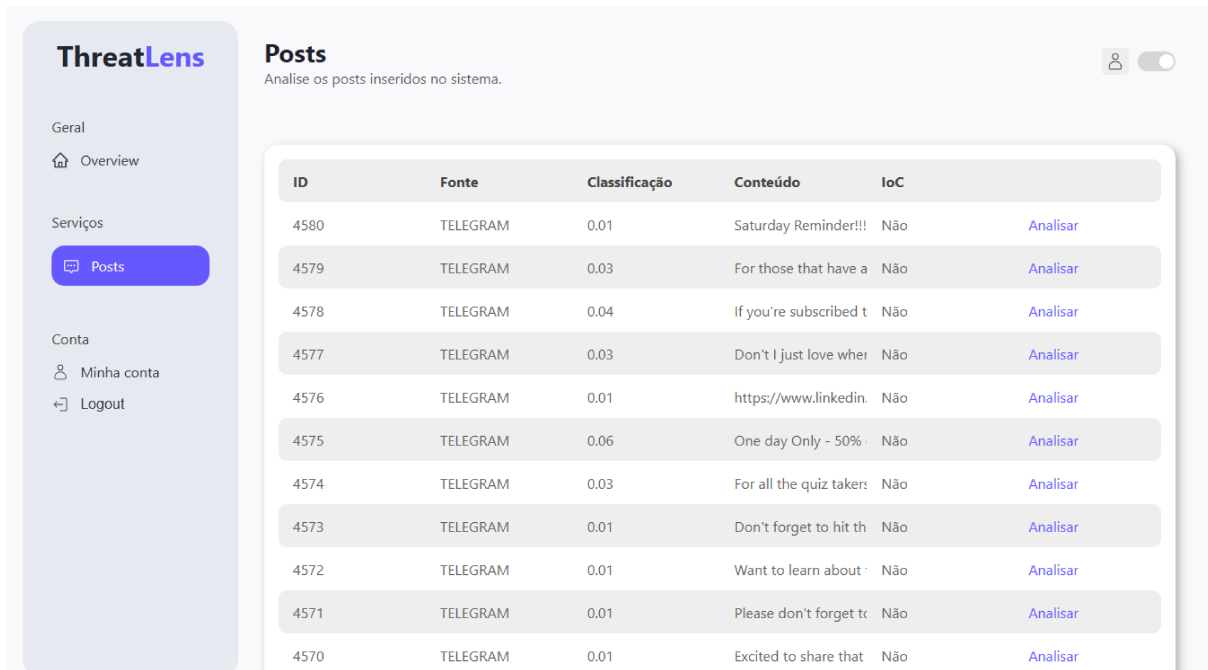


Fonte: Autoria própria.

Após realizar o login, o usuário é direcionado para o *dashboard* principal da aplicação, apresentado nas Figuras 4 e 5. Essa página reúne informações consolidadas sobre os dados processados pelo sistema de detecção de ameaças cibernéticas, apresentando indicadores, gráficos estatísticos, fontes monitoradas e uma nuvem de palavras. A tela também possui filtros temporais, permitindo analisar os dados considerando diferentes períodos, como dia, semana, mês ou ano.

A página de publicações, exibida na Figura 6, permite consultar os dados coletados pelo sistema de detecção de ameaças cibernéticas em formato de tabela. Para facilitar a visualização de grandes quantidades de registros, foi implementada paginação, permitindo navegar entre diferentes conjuntos de resultados.

Figura 6 – Tela de publicações analisadas



ID	Fonte	Classificação	Conteúdo	IoC	
4580	TELEGRAM	0.01	Saturday Reminder!!!	Não	Analisar
4579	TELEGRAM	0.03	For those that have a	Não	Analisar
4578	TELEGRAM	0.04	If you're subscribed t	Não	Analisar
4577	TELEGRAM	0.03	Don't I just love wher	Não	Analisar
4576	TELEGRAM	0.01	https://www.linkedin.	Não	Analisar
4575	TELEGRAM	0.06	One day Only - 50%	Não	Analisar
4574	TELEGRAM	0.03	For all the quiz taker:	Não	Analisar
4573	TELEGRAM	0.01	Don't forget to hit th	Não	Analisar
4572	TELEGRAM	0.01	Want to learn about	Não	Analisar
4571	TELEGRAM	0.01	Please don't forget to	Não	Analisar
4570	TELEGRAM	0.01	Excited to share that	Não	Analisar

Fonte: Autoria própria.

A página de conta do usuário, apresentada na Figura 7, concentra as informações relacionadas ao perfil autenticado. O nome de usuário e o e-mail são exibidos apenas para consulta, enquanto a senha pode ser alterada por meio de um modal específico, sem sair da página atual.

Figura 7 – Tela Minha conta

ThreatLens

Minha conta
Gerencie as informações da sua conta

Geral

Overview

Serviços

Posts

Conta

Minha conta

Logout

Nome de usuário

pedro

Email

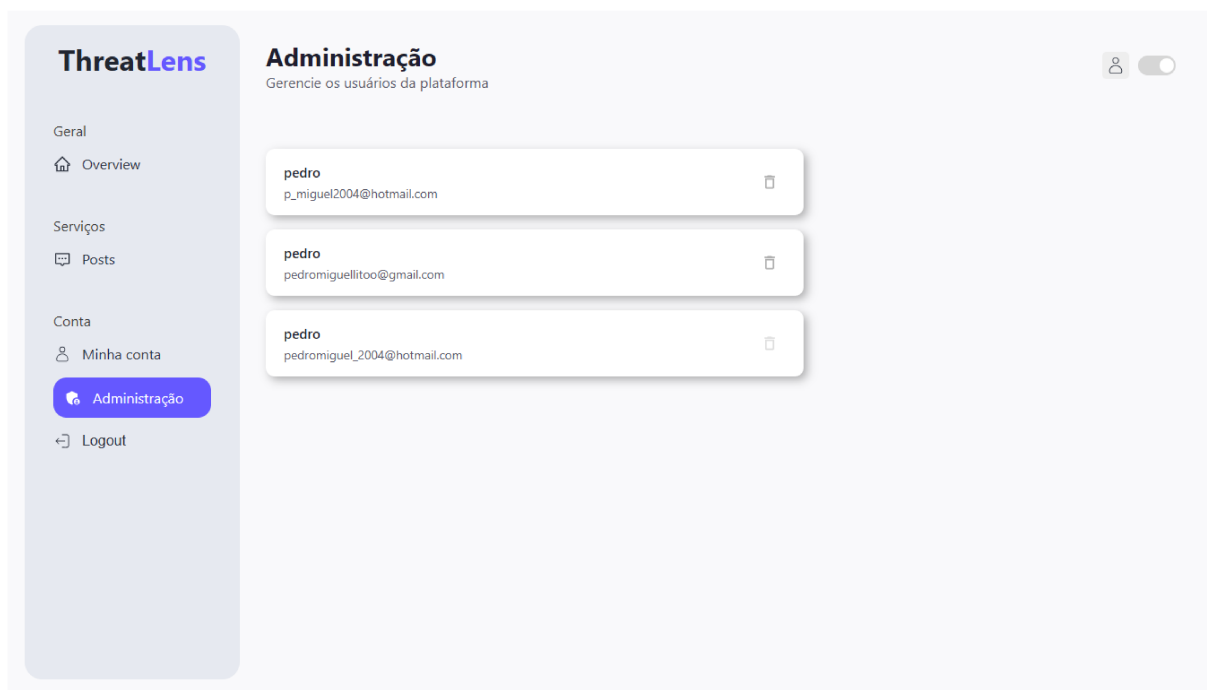
pedromiguel_2004@hotmail.com

Senha

Fonte: Autoria própria.

A aplicação também possui uma área administrativa destinada ao gerenciamento de usuários, conforme mostrado na Figura 8. Essa tela permite consultar os usuários cadastrados e realizar operações administrativas, como exclusão de contas. Para evitar alterações acidentais, ações críticas exigem confirmação antes de serem executadas.

Figura 8 – Tela administrativa



Fonte: Autoria própria.

Por fim, todas as páginas foram desenvolvidas utilizando CSS Modules, mantendo os estilos isolados entre componentes e evitando conflitos de regras CSS. A responsividade foi considerada durante a implementação, com adaptações no menu de navegação, tabelas e formulários para diferentes tamanhos de tela. Dessa forma, a plataforma mantém sua usabilidade em dispositivos com diferentes resoluções.

6.3.5 Integração com o *back-end*

Após a implementação da estrutura visual da aplicação, foi realizada a integração com o *back-end* responsável pelo processamento e disponibilização dos dados utilizados pela plataforma ThreatLens. Essa etapa teve como objetivo permitir que as informações exibidas na interface fossem obtidas dinamicamente a partir dos serviços disponibilizados pela API, substituindo dados estáticos utilizados durante o desenvolvimento inicial.

Para manter a separação entre apresentação e comunicação com o servidor, todas as requisições foram centralizadas no diretório `/services`, conforme a arquitetura definida anteriormente. Essa abordagem evitou que componentes e páginas realizassem chamadas diretamente à API, favorecendo a organização do código e facilitando futuras manutenções.

A comunicação entre o *front-end* e o *back-end* foi realizada por meio de requisições HTTP aos endpoints disponibilizados pelo servidor. Entre as principais operações

implementadas destacam-se:

- autenticação de usuários
- recuperação de dados do *dashboard*
- consulta de publicações analisadas
- operações administrativas relacionadas aos usuários da plataforma.

Os códigos 6.1 e 6.2 mostram dois exemplos de implementações de serviços, chamados `PostsService` e `AdminService`, responsáveis pela comunicação com os endpoints de publicações e administração da API, respectivamente.

Código 6.1 – Serviço Posts Service responsável pela comunicação com os endpoints de publicações da API

```
1 import api from "../api";
2
3 export async function getPosts (params = {}) {
4     const { data } = await api.get("/posts", { params });
5     return data;
6 }
7
8 export async function getStats (params = {}) {
9     const { data } = await api.get("/posts/stats", { params });
10    return data;
11 }
```

Código 6.2 – Serviço Admin Service responsável pela comunicação com os endpoints de administração da API

```
1 import api from "../api";
2
3 export const adminService = {
4     listAll: ({ role, search, page = 0, size = 20 } = {}) =>
5         api.get("/admin/users", { params: { role, search, page,
6             size, sort: "username,asc" } }),
7     deleteUser: (id) => api.delete(`/admin/users/${id}`),
8 };
```

Os dados retornados pelos serviços não são consumidos diretamente pelos componentes da interface, mas por meio dos hooks personalizados definidos no diretório `/hooks`,

como `usePosts`, `useStats` e `useUsers`. Cada hook encapsula a chamada ao serviço correspondente e gerencia o estado da requisição, incluindo o carregamento dos dados, o tratamento de erros e a disponibilização do resultado já formatado para os componentes.

Optou-se pela implementação dessa camada adicional pensando em reforçar a separação de responsabilidades estabelecida na arquitetura do projeto, isolando a lógica de obtenção e gerenciamento de estado dos dados da camada puramente visual da aplicação.

7 Avaliação

Este capítulo apresenta a avaliação do *front-end* da plataforma ThreatLens, com o objetivo de demonstrar o atendimento aos requisitos funcionais (RF) e não funcionais (RNF) definidos na fase de projeto.

7.1 Procedimentos de avaliação

A validação do sistema foi conduzida por meio da simulação do fluxo de uso da aplicação. Os testes foram executados em ambiente de desenvolvimento local, acompanhando a jornada de um usuário desde o acesso inicial até a exploração dos dados consolidados, verificando na prática o comportamento da interface e a comunicação com o *back-end*.

7.2 Avaliação do fluxo de uso e requisitos

A demonstração do cumprimento dos requisitos é detalhada a seguir, seguindo a ordem natural de navegação do sistema.

7.2.1 Acesso e Autenticação

O primeiro contato com a plataforma valida os requisitos de controle de acesso (RNF02). A tela de login (apresentada anteriormente na Figura 2) permite a autenticação segura de usuários cadastrados (RF02). Caso necessário, a partir dela o usuário pode iniciar o fluxo de cadastro (RF01) ou solicitar a redefinição de sua senha (RF03). Para garantir a segurança dessas operações, o sistema exige a confirmação de identidade por meio de um código enviado por e-mail, cuja interface de validação foi ilustrada na Figura 3. O encerramento da sessão (RF04) é garantido por meio do botão de *logout* presente na sidebar de todas as telas internas.

7.2.2 Visão Geral (*Dashboard*)

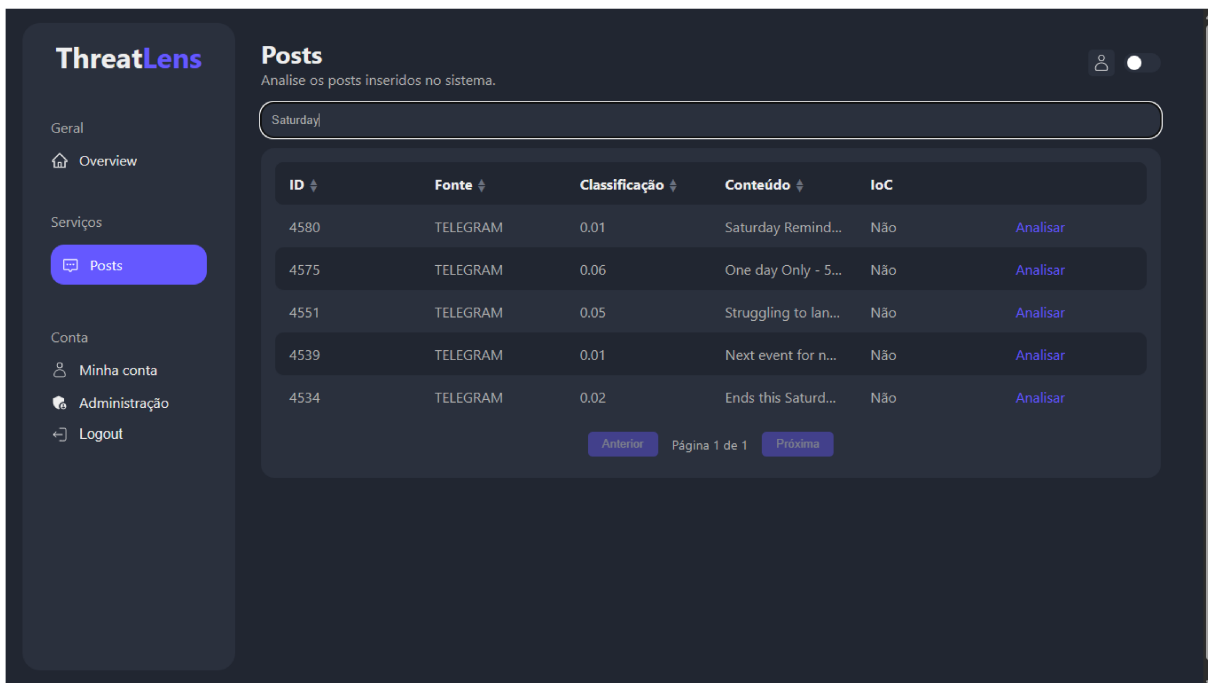
Após o login, o usuário visualiza o *dashboard* (Figura 4), que atende à necessidade de exibir um painel consolidado sobre as ameaças (RF09). A tela atende aos critérios de usabilidade (RNF03) ao organizar os indicadores em gráficos interativos e em uma nuvem de palavras (Figura 5), facilitando a leitura rápida dos dados.

7.2.3 Consulta de Publicações

O detalhamento das ameaças ocorre na aba de publicações. Inicialmente, o sistema carrega com sucesso os dados brutos provenientes do *back-end* (RF06), conforme exibido anteriormente na Figura 6.

Para atestar a interatividade da interface na localização de informações específicas, a Figura 9 demonstra o funcionamento da busca textual (RF07), em que a tabela exibe instantaneamente apenas os registros que correspondem ao termo pesquisado.

Figura 9 – Filtragem ativa de publicações por meio de busca textual



The screenshot shows the ThreatLens application interface. On the left is a dark sidebar with the 'ThreatLens' logo and navigation links: 'Geral', 'Overview', 'Serviços' (with 'Posts' highlighted), and 'Conta' (with 'Minha conta', 'Administração', and 'Logout'). The main area is titled 'Posts' with the subtitle 'Analise os posts inseridos no sistema.' and a search bar containing 'Saturday'. Below the search bar is a table with 5 columns: 'ID', 'Fonte', 'Classificação', 'Conteúdo', and 'IoC'. The table displays 5 rows of data, all from 'TELEGRAM'. Each row has an 'Analisar' button. At the bottom of the table are 'Anterior', 'Página 1 de 1', and 'Próxima' buttons.

ID	Fonte	Classificação	Conteúdo	IoC	
4580	TELEGRAM	0.01	Saturday Remind...	Não	Analisar
4575	TELEGRAM	0.06	One day Only - 5...	Não	Analisar
4551	TELEGRAM	0.05	Struggling to lan...	Não	Analisar
4539	TELEGRAM	0.01	Next event for n...	Não	Analisar
4534	TELEGRAM	0.02	Ends this Saturd...	Não	Analisar

Fonte: Autoria própria.

Além da busca, a plataforma permite organizar a visualização dos dados estruturados. As Figuras 10 e 11 ilustram a ordenação dinâmica dos registros (RF08), reorganizando os itens de acordo com atributos específicos, como a classificação. É importante destacar que toda a navegação entre as páginas e a aplicação desses filtros ocorrem de forma assíncrona, cumprindo o requisito de transição sem recarregamento completo da página (RF10).

Figura 10 – Publicações ordenadas pela classificação em ordem decrescente

ThreatLens

Geral

Overview

Serviços

Posts

Conta

Minha conta

Administração

Logout

Posts

Analise os posts inseridos no sistema.

ID	Fonte	Classificação	Conteúdo	IoC	
3881	TELEGRAM	0.99	NTLM Hash ...	Não	Analisar
3992	TELEGRAM	0.96	Ransomware Att...	Não	Analisar
4014	TELEGRAM	0.94	CyberWeekly...	Não	Analisar
4483	TELEGRAM	0.93	Extremely...	Não	Analisar
3822	TELEGRAM	0.92	FBI: Cybercrí...	Não	Analisar
4153	TELEGRAM	0.92	EvilLoader: Yeste...	Não	Analisar
4010	TELEGRAM	0.91	Pokemon Develo...	Não	Analisar
3957	TELEGRAM	0.91	Cyble Sensors D...	Não	Analisar
3854	TELEGRAM	0.91	Iran-Linked ...	Não	Analisar
4453	TELEGRAM	0.90	Cisco ASA an...	Não	Analisar
3873	TELEGRAM	0.90	APT29 Deplo...	Não	Analisar

Fonte: Autoria própria.

Figura 11 – Publicações ordenadas pela classificação em ordem crescente

ThreatLens

Geral

Overview

Serviços

Posts

Conta

Minha conta

Administração

Logout

Posts

Analise os posts inseridos no sistema.

ID	Fonte	Classificação	Conteúdo	IoC	
4036	TELEGRAM	0.01	SPECIAL OFFER ...	Não	Analisar
4025	TELEGRAM	0.01	SPECIAL OFFER ...	Não	Analisar
4044	TELEGRAM	0.01	SPECIAL OFFER ...	Não	Analisar
3874	TELEGRAM	0.01	If Boards Do...	Não	Analisar
4139	TELEGRAM	0.01	PJobRAT makes ...	Não	Analisar
4030	TELEGRAM	0.01	Every year it bec...	Não	Analisar
4049	TELEGRAM	0.01	Every year it bec...	Não	Analisar
4027	TELEGRAM	0.01	Every year it bec...	Não	Analisar
4096	TELEGRAM	0.01	Every year it bec...	Não	Analisar
4507	TELEGRAM	0.01	These ones are g...	Não	Analisar
4545	TELEGRAM	0.01	Finding it hard t...	Não	Analisar

Fonte: Autoria própria.

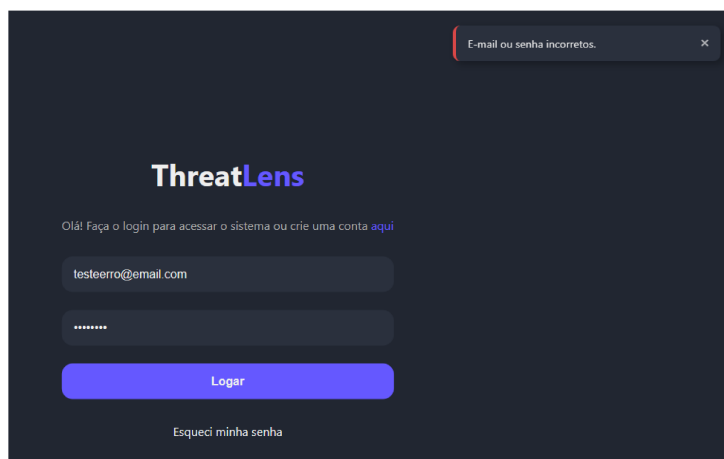
7.2.4 Gerenciamento de Contas

Na seção "Minha Conta" (Figura 7), o usuário autenticado pode consultar seu nome de usuário e e-mail e alterar sua senha (RF05). A edição do nome de usuário e do e-mail não está disponível nessa tela. Além disso, usuários com perfil de administrador possuem acesso à tela de gestão (Figura 8), onde podem visualizar e remover contas cadastradas na plataforma (RF12).

7.2.5 Feedback Visual

Para garantir que o usuário compreenda o resultado de suas ações, o sistema implementa mensagens de *feedback* (RF11). A Figura 12 apresenta uma tentativa de autenticação malsucedida, na qual a interface exibe uma notificação informando que o e-mail ou a senha estão incorretos.

Figura 12 – Mensagem de erro de autenticação por e-mail ou senha incorretos



Fonte: Autoria própria.

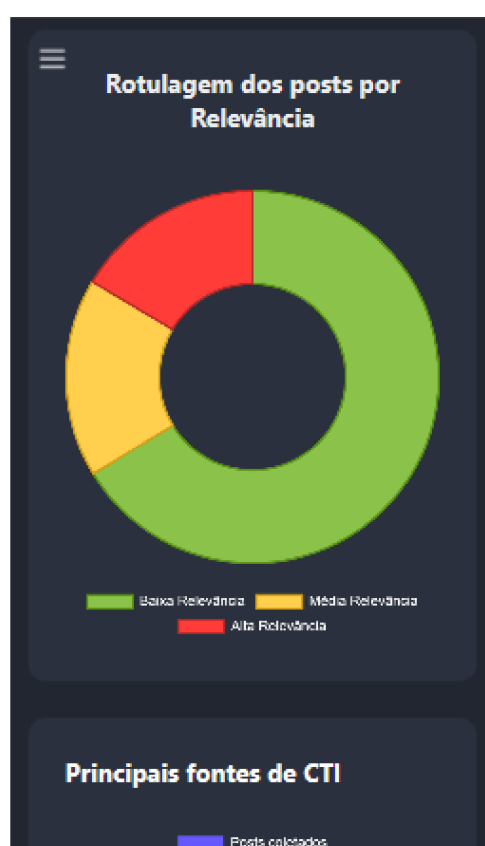
7.2.6 Responsividade e Alternância de Tema

A validação estrutural comprovou que o *layout* atende ao requisito de responsividade (RNF01). A Figura 13 apresenta os indicadores e a tabela de publicações mais relevantes do *dashboard* em uma tela de dispositivo móvel. A Figura 14 mostra o gráfico de relevância das publicações, enquanto a Figura 15 exibe o gráfico das fontes de CTI e a nuvem de palavras nessa mesma configuração de tela.

Figura 13 – Indicadores e publicações mais relevantes do *dashboard* em dispositivo móvel

Fonte: Autoria própria.

Figura 14 – Gráfico de relevância das publicações em dispositivo móvel



Fonte: Autoria própria.

Figura 15 – Gráfico das fontes de CTI e nuvem de palavras em dispositivo móvel



Fonte: Autoria própria.

O sistema também atende ao requisito de alternância de tema (RF13). A Figura 16 exibe a interface com o modo escuro ativado, evidenciando o funcionamento dinâmico das variáveis CSS para a troca da paleta de cores.

Figura 16 – Plataforma com o tema escuro (*Dark Mode*) ativado



Fonte: Autoria própria.

7.2.7 Manutenibilidade

Por fim, o requisito de manutenibilidade (RNF04) é garantido de forma indireta pela arquitetura adotada. A divisão do código em componentes, serviços e *hooks*, aliada ao uso do ESLint, assegura uma base de código modularizada, o que facilitará a correção de falhas e a implementação de novos recursos.

7.3 Resultados da avaliação

A validação demonstrou que a plataforma ThreatLens atende a todos os requisitos delineados. O fluxo contínuo de operações na interface ocorreu sem falhas de roteamento ou de apresentação. Com os resultados obtidos, conclui-se que o *front-end* é plenamente capaz de suportar a camada visual do projeto de pesquisa, fornecendo aos usuários uma ferramenta ágil, responsiva e segura para a análise de dados cibernéticos.

8 Conclusão

O crescimento das ameaças cibernéticas e o aumento do volume de informações provenientes de diferentes fontes tornam cada vez mais necessária a utilização de ferramentas capazes de organizar, interpretar e apresentar esses dados de forma clara e eficiente. Nesse contexto, este trabalho teve como objetivo desenvolver o *front-end* da plataforma ThreatLens, uma aplicação web voltada à visualização e análise de dados produzidos por um sistema de detecção de ameaças cibernéticas, contribuindo para apoiar profissionais da área na interpretação de informações relevantes para a tomada de decisões.

Os objetivos definidos foram alcançados por meio da implementação de uma interface web responsiva, interativa e baseada em componentes reutilizáveis, capaz de integrar-se ao *back-end* da plataforma para consumir os dados. Durante o desenvolvimento, foram realizadas as etapas de levantamento de requisitos, prototipação da interface, definição da arquitetura da aplicação e implementação das páginas, componentes e serviços responsáveis pela comunicação com a API. Como resultado, foi obtida uma aplicação organizada, modular e alinhada às boas práticas de desenvolvimento de software, favorecendo sua manutenção e evolução.

Entre as principais contribuições deste trabalho destaca-se o desenvolvimento de uma camada de visualização capaz de transformar dados provenientes de diferentes fontes em informações apresentadas de maneira intuitiva por meio de gráficos, indicadores e tabelas interativas. Além disso, a arquitetura adotada, baseada em componentização e separação de responsabilidades, contribui para a escalabilidade da aplicação e facilita a incorporação de novas funcionalidades. O sistema desenvolvido também consolida os resultados obtidos por pesquisas anteriores do grupo de pesquisa, integrando a etapa de visualização aos mecanismos de coleta, classificação e processamento de dados já existentes.

Como trabalhos futuros, podem ser exploradas funcionalidades que ampliem o potencial analítico da plataforma. Entre elas destacam-se a implementação de notificações em tempo real sobre novas ameaças identificadas, a criação de filtros avançados para consultas mais específicas, a exportação de relatórios em diferentes formatos, a disponibilização de novos tipos de visualizações e métricas estatísticas, a integração com novas fontes de inteligência de ameaças e a realização de avaliações de usabilidade com especialistas em cibersegurança para identificar oportunidades de melhoria na experiência do usuário.

APÊNDICE A – Declaração de uso de IA generativa

Para a produção deste trabalho, foi utilizado o ChatGPT (OpenAI, versão GPT-5.5-mini, acesso em agosto de 2026) para auxílio na revisão gramatical, aprimoramento da redação acadêmica e organização da estrutura textual.

O uso da ferramenta ocorreu durante as etapas de revisão da escrita, adequação da linguagem acadêmica e estruturação de ideias para auxiliar na elaboração de alguns trechos do documento. A ferramenta também foi utilizada como apoio na organização inicial de conteúdos, contribuindo para o aprimoramento da clareza e coerência textual. Todo conteúdo produzido com auxílio da ferramenta foi analisado, revisado e adaptado criticamente pelo autor, que assume integral responsabilidade pelo trabalho final.

Referências

- ANDERSON, R. *Security Engineering: A Guide to Building Dependable Distributed Systems*. 3. ed. Hoboken, NJ: Wiley, 2020. ISBN 9781119642787. 15
- BARTLETT, J. *The Dark Net: Inside the Digital Underworld*. New York: Melville House, 2015. ISBN 9781612194899. 18
- BASHEER, R.; ALKHATIB, B. Threats from the dark: a review over dark web investigation research for cyber threat intelligence. *Journal of Computer Networks and Communications*, v. 2021, p. 1–21, 2021. 17, 18, 19
- BERGMAN, M. K. *The Deep Web: Surfacing Hidden Value*. 2001. White Paper, BrightPlanet. 18
- BÖHM, F.; MENGES, F.; PERNUL, G. Graph-based visual analytics for cyber threat intelligence. *Cybersecurity*, SpringerOpen, v. 1, n. 1, p. 16, 2018. 20, 21
- CAMILLERI, J. *Surface Web, Deep Web and Dark Web: Three Levels of Exploitation*. 2020. Preprint disponibilizado no ResearchGate. Disponível em: <https://www.researchgate.net/publication/347986866_Surface_Web_Deep_Web_and_Dark_Web_three_levels_of_exploitation>. Acesso em: 7 set. 2026. 18
- CHACON, S.; STRAUB, B. *Pro Git*. 2. ed. Nova York: Apress, 2014. Disponível em: <<https://git-scm.com/book/en/v2>>. Acesso em: 16 mai. 2026. 27
- CHART.JS. *Chart.js documentation*. 2026. Disponível em: <<https://www.chartjs.org/docs/>>. Acesso em: 16 mai. 2026. 27
- CYBERSECURITY VENTURES. *Cybersecurity Ventures Report on Cybercrime*. 2023. Disponível em: <<https://cybersecurityventures.com/>>. Acesso em: 16 mai. 2026. 11
- CYBLE. *Cyber Threat Intelligence 2025*. 2025. Disponível em: <<https://cyble.com/knowledge-hub/cyber-threat-intelligence-2025/>>. Acesso em: 16 mai. 2026. 17
- DALZIEL, H. *How to Define and Build an Effective Cyber Threat Intelligence Capability*. Waltham, MA: Syngress, 2014. 16
- DATAREPORTAL. *Digital 2025: Global Overview Report*. 2025. Disponível em: <<https://datareportal.com/reports/digital-2025-global-overview-report>>. Acesso em: 16 mai. 2026. 11
- DAVIES, V. *The history of cybersecurity*. 2021. Cyber Magazine. 15
- FIGMA. *Figma Help Center*. 2026. Disponível em: <<https://help.figma.com>>. Acesso em: 16 mai. 2026. 25
- GITHUB. *GitHub Docs*. 2026. Disponível em: <<https://docs.github.com>>. Acesso em: 16 mai. 2026. 27

- GROUP-IB. *What is the Deep Web: the hidden Internet*. 2025. Disponível em: <<https://www.group-ib.com/resources/knowledge-hub/deep-web/>>. Acesso em: 16 mai. 2026. 18
- IBM. *O que é cibersegurança?* 2024a. Disponível em: <<https://www.ibm.com/br-pt/topics/cybersecurity>>. 16
- IBM. *Tipos de ameaças cibernéticas*. 2024b. Disponível em: <<https://www.ibm.com/br-pt/think/topics/cyberthreats-types>>. 14
- IKLODY, A. et al. *Decaying Indicators of Compromise*. 2018. Disponível em: <<https://arxiv.org/abs/1803.11052>>. 17
- JESUS FILHO, S. A. d. *Identificação de posts maliciosos na dark web utilizando Aprendizado de Máquina Supervisionado*. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 2023. Disponível em: <<https://repositorio.ufu.br/handle/123456789/41232>>. 11, 20, 21
- JIANG, L. et al. Efficient deep web crawling using reinforcement learning. In: ZAKI, M. J. et al. (Ed.). *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer, 2010, (Lecture Notes in Computer Science, v. 6118). p. 428–439. 18
- KATUMBA, S.; COETZEE, S. Employing search engine optimization (seo) techniques for improving the discovery of geospatial resources on the web. *ISPRS International Journal of Geo-Information*, v. 6, n. 9, p. 284, 2017. 18
- KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: Uma Abordagem Top-Down*. 6. ed. São Paulo: Pearson Education do Brasil, 2013. 11
- LUGHBI, H.; MARS, M.; ALMOTAIRI, K. A novel nlp-driven dashboard for interactive cyberattacks tweet classification and visualization. *Information*, MDPI, v. 15, n. 3, p. 137, 2024. 20, 21
- MCMILLAN, R. *Definition: Threat Intelligence*. 2013. Gartner. Disponível em: <<https://www.gartner.com/en/documents/2487216>>. 11
- META. *React: a JavaScript library for building user interfaces*. 2024. Disponível em: <<https://react.dev>>. Acesso em: 16 mai. 2026. 25
- MICROSOFT. *O que é um ciberataque?* 2024. Disponível em: <<https://www.microsoft.com/pt-br/security/business/security-101/what-is-a-cyberattack>>. Acesso em: 16 mai. 2026. 15
- MICROSOFT. *Visual Studio Code Documentation*. 2026. Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 16 mai. 2026. 28
- PROXIMONIVEL. *Surface web, deep web e dark web: qual a diferença?* 2025. Disponível em: <<https://proximonivel.embratel.com.br/conheca-a-diferenca-entre-surface-web-deep-web-e-dark-web/>>. Acesso em: 16 mai. 2026. 18
- RAHMAN, M. R.; MAHDAVI-HEZAVEH, R.; WILLIAMS, L. What are the attackers doing now? automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: A survey. *arXiv preprint*, v. 2109.06808v1, 2021. Disponível em: <<https://arxiv.org/abs/2109.06808>>. 16

- RANKING ARTICLES. *The Evolution of Content*. 2023. Disponível em: <<https://ranking-articles.com/the-evolution-of-content/>>. Acesso em: 16 mai. 2026. 18
- REACT. *React Reference Docs*. 2024. Disponível em: <<https://react.dev/reference/react>>. Acesso em: 16 mai. 2026. 26
- REACT ROUTER. *React Router Documentation*. 2024. Disponível em: <<https://reactrouter.com>>. Acesso em: 16 mai. 2026. 26
- RED HAT. *Casos de uso para IA em cibersegurança*. 2024. Disponível em: <<https://www.redhat.com/pt-br/blog/use-cases-ai-cybersecurity>>. Acesso em: 16 mai. 2026. 16
- REIS, A. L. F. *Análise de posts maliciosos na Dark Web usando aprendizado de máquina não supervisionado*. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2024. Disponível em: <<https://repositorio.ufu.br/handle/123456789/44118>>. 20, 21
- SCHILIRO, F. *Towards a Contemporary Definition of Cybersecurity*. 2022. Draft report, Macquarie University. 15
- STALLINGS, W.; BROWN, L. *Computer Security: Principles and Practice*. 2. ed. Upper Saddle River, NJ: Prentice Hall, 2012. ISBN 978-0-13-277506-9. 14
- VIEIRA, C. R. *Desenvolvimento de um bot de captura de mensagens do Telegram para inteligência de ameaças cibernéticas*. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2025. Disponível em: <<https://repositorio.ufu.br/handle/123456789/45319>>. 20, 21
- VITE. *Vite: Next Generation Frontend Tooling*. 2026. Disponível em: <<https://vite.dev>>. Acesso em: 16 mai. 2026. 28
- W3C. *Cascading Style Sheets*. 2021. Disponível em: <<https://www.w3.org/Style/CSS/>>. Acesso em: 16 mai. 2026. 26
- WAZLAWICK, R. S. *Metodologia de Pesquisa para Ciência da Computação*. 3. ed. Rio de Janeiro: GEN LTC, 2020. 12