Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus João Pessoa

Programa de Pós-Graduação em Tecnologia da Informação

Nível Mestrado Profissional

YURI FEITOSA NEGÓCIO

THREAT COPILOT: UM SISTEMA DE RECOMENDAÇÃO PARA MODELAGEM DE AMEAÇAS DE SOFTWARE

DISSERTAÇÃO DE MESTRADO

JOÃO PESSOA

2026

YURI FEITOSA NEGÓCIO

THREAT COPILOT: UM SISTEMA DE RECOMENDAÇÃO PARA MODELAGEM DE AMEAÇAS DE SOFTWARE

Dissertação apresentada como requisito parcial para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB.

Orientador: Prof. Dra. Juliana Dantas Ribeiro Viana de Medeiros
Coorientador: Prof. Dra. Nadja da Nóbrega Rodrigues

João Pessoa

2026

Dados Internacionais de Catalogação na Publicação – CIP
Biblioteca Nilo Peçanha – IFPB, *Campus* João Pessoa

N384t Negócio, Yuri Feitosa.
Threat Copilot : um sistema de recomendação para modelagem de ameaças de software / Yuri Feitosa Negócio. – 2026.
141 f. : il.

Dissertação (Mestrado em Tecnologia da Informação) – Instituto Federal da Paraíba – IFPB / Programa de Pós-Graduação em Tecnologia da Informação - PPGTI.
Orientadora: Profa. Dra. Juliana Dantas Ribeiro Viana de Medeiros.
Coorientadora: Profa. Dra. Nadja da Nóbrega Rodrigues.

1. Modelagem de ameaças. 2. Sistema de recomendação. 3. Segurança de software. I. Título.

CDU 004.056

Bibliotecário responsável Marx da Silva Medeiros – CRB15/470

YURI FEITOSA NEGÓCIO

THREAT COPILOT: UM SISTEMA DE RECOMENDAÇÃO PARA MODELAGEM DE AMEAÇAS

DISSERTAÇÃO apresentada como requisito para obtenção do título de Mestre em Tecnologia da Informação, pelo Programa de Pós- Graduação em Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - IFPB - Campus João Pessoa.

Aprovado em 27 de fevereiro de 2026.

Membros da Banca Examinadora:

Prof. Dr. JULIANA DANTAS RIBEIRO VIANA DE MEDEIROS
Instituto Federal da Paraíba (IFPB)
Orientadora

Prof. Dr. NADJA DA NÓBREGA RODRIGUES
Instituto Federal da Paraíba (IFPB)
Examinador

Prof. Dr. FRANCISCO PETRONIO ALENCAR DE MEDEIROS
Instituto Federal da Paraíba (IFPB)
Examinador

Prof. Dr. EDNALDO DILORENZO DE SOUZA FILHO
Instituto Federal da Paraíba (IFPB)
Examinador

Documento assinado eletronicamente por:

- **Juliana Dantas Ribeiro Viana de Medeiros, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 27/02/2026 18:36:23.
- **Francisco Petronio Alencar de Medeiros, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 27/02/2026 20:18:19.
- **Nadja da Nobrega Rodrigues**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 28/02/2026 07:14:26.
- **Ednaldo Dilorenzo de Souza Filho**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 22/04/2026 17:31:57.

Este documento foi emitido pelo SUAP em 19/02/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 836770
Verificador: 4f2ccdee7b
Código de Autenticação:

"Temos todos nossas máquinas do tempo, não? As que nos levam para trás são nossas memórias. As que nos levam à frente são nossos sonhos."— H.G. Wells

AGRADECIMENTOS

Agradeço a Deus, a Jesus, à Santíssima Trindade, à Virgem Maria e a todos os santos que sempre estiveram presentes em minha vida e tornaram possível a concretização deste sonho tão antigo. É bom confiar! Mateus 6:19-34.

À minha amada esposa Renata Maurera, fortaleza nos momentos mais difíceis, que com amor e paciência supriu minha ausência junto aos nossos filhos, incentivando-me e tornando possível a realização deste sonho. Aos meus queridos filhos Arthur e Maria Eduarda, que sempre me apoiaram com carinho e compreensão. Que o meu exemplo possa edificar e inspirar suas vidas!

À minha mãe Célia, a verdadeira "miolos dourados", por me ensinar o valor do esforço, da dedicação e da fé. Ao meu pai Cláudio, hoje fisicamente ausente, mas eternamente presente em minhas memórias, pensamentos e intenções. Às minhas irmãs Denise, Dayse e Claudia Rossana, que sempre foram pilares de apoio, oração e incentivo em minha caminhada.

À minha família — Dra. Aldeiza, José Jesus, Rebeca, Yanko, Lula, Manu, Fred, André Lucena e André Torquato — pelo apoio constante, pelas palavras de encorajamento e pelos gestos de carinho que me motivaram a seguir em frente.

Aos amigos do trabalho, do aquário da segurança e do ECC de Neves, pelo incentivo, pelas palavras de conforto e pela compreensão durante o período de estudos. Aos amigos pessoais, que sempre acreditaram em minha capacidade e me lembraram de não desistir. Aos colegas do mestrado, com quem compartilhei aprendizados, desafios e momentos de amizade e admiração.

Aos amigos de infância e da vida, em especial aos essenciais para esta realização Nilton Trindade, Loreno Feitosa, Vinicius Domingues, Felipe Crisanto, Thiago Prota, Victor Gouveia e Karla Roberta obrigado pelo apoio incondicional nos momentos mais críticos, quando a força deles tornou possível a continuidade desta jornada.

Aos professores e amigos, cuja sabedoria e amor pela pesquisa foram exemplo, em especial a Glêdson Elias e ao eterno amigo Gustavo Cavalcanti (in memoriam). As profissionais Kaline e Tida que auxiliaram nos cuidados dos meus filhos e da minha casa diante de dias tão difíceis. Que o exemplo de esforço acadêmico seja luz para suas famílias.

E, de modo muito especial, aos professores do Programa de Pós-Graduação em Tecnologia da Informação do Instituto Federal da Paraíba (IFPB), em especial à minha orientadora, Dra. Juliana Dantas Ribeiro de Viana Medeiros e à Dra. Nadja da Nóbrega Rodrigues, pela dedicação, paciência, confiança e constante incentivo, tornando este caminho mais leve e possível de ser trilhado.

RESUMO

Os processos de desenvolvimento seguro de software buscam garantir, por meio da adoção de práticas, que a construção do software resulte em produtos capazes de funcionar adequadamente, mesmo diante de ataques. Uma das atividades mais relevantes em um ciclo de desenvolvimento seguro é a identificação antecipada das falhas de segurança por meio da modelagem de ameaças. Diversos métodos para modelagem de ameaças foram propostos, tanto na indústria quanto na academia. Mesmo assim, a execução dessa atividade ainda não tem sido trivial para os times de desenvolvimento. Alguns desafios persistem, principalmente devido ao tempo necessário para realizar a atividade e à exigência de profissionais especializados em modelagem de ameaças. Nesse contexto, reutilizar o conhecimento de especialistas e permitir que a identificação das ameaças de segurança seja relevante e contínua é essencial para reduzir o tempo de elaboração dos modelos de ameaças. Este trabalho teve como objetivo geral, propor e desenvolver um sistema de recomendação capaz de automatizar a etapa de elicitação de um processo de modelagem de ameaças de software a partir do reuso de conhecimento proveniente de modelos de ameaças previamente elaborados em uma organização. O Sistema, denominado *Threat Copilot*, compara etiquetas, elementos e relacionamentos de diagramas de fluxo de dados por meio de técnicas de similaridade semântica apoiadas em métricas derivadas de uma WordNet específica para o domínio da modelagem de ameaças. Um conjunto de modelos reais foi estruturado e utilizado para avaliar o desempenho do sistema, considerando métricas, como *precision*, *recall* e *f1-measure*. Adicionalmente, foi conduzida uma avaliação centrada no usuário, fundamentada no modelo integrado TAM-TTF, com o objetivo de analisar a utilidade percebida, a facilidade de uso, o ajuste tarefa–tecnologia e a intenção de uso da ferramenta no contexto organizacional. Os resultados revelaram que o sistema é capaz de recomendar ameaças relevantes, apresentando desempenho de 51%, 72% e 56% para *precision*, *recall* e *f1-measure* na avaliação *offline* e 60%, 75% e 67% na avaliação *online*, respectivamente. Esses achados indicam um potencial para redução de esforço na elicitação de ameaças e maior consistência no processo de modelagem. De forma geral, os dados apontam para uma percepção favorável quanto ao uso da ferramenta na organização avaliada. Entretanto, os resultados obtidos são limitados aos 34 de modelos de ameaças disponíveis como base de conhecimento. A utilização futura de mais modelos de ameaças e a inclusão de novas características dos elementos dos modelos de ameaças podem aprimorar a qualidade das recomendações geradas.

Palavras-chaves: modelagem de ameaças; sistema de recomendação; segurança de software.

ABSTRACT

Secure software development processes aim to ensure, through the adoption of practices, that software construction results in products capable of operating properly even under attack. One of the most relevant activities in a secure development lifecycle is the early identification of security flaws through threat modeling. Several threat modeling methods have been proposed in both industry and academia. Even so, performing this activity has not been trivial for development teams. Some challenges persist, mainly due to the time required to carry out the activity and the need for professionals specialized in threat modeling. In this context, reusing experts' knowledge and enabling the identification of security threats to be relevant and continuous is essential to reduce the time required to produce threat models. The overall objective of this work was to propose and develop a recommender system capable of automating the elicitation stage of a software threat modeling process by reusing knowledge derived from threat models previously created within an organization. The system, named Threat Copilot, compares labels, elements, and relationships of data flow diagrams using semantic similarity techniques supported by metrics derived from a WordNet tailored to the threat modeling domain. A set of real-world models was structured and used to evaluate the system's performance using metrics such as precision, recall, and F1-measure. In addition, a user-centered evaluation grounded in the integrated TAM–TTF model was conducted to analyze perceived usefulness, ease of use, task–technology fit, and intention to use the tool in the organizational context. The results showed that the system can recommend relevant threats, achieving 51%, 72%, and 56% for precision, recall, and F1-measure in the offline evaluation, and 60%, 75%, and 67% in the online evaluation, respectively. These findings indicate potential for reducing effort in threat elicitation and increasing consistency in the modeling process. Overall, the data point to a favorable perception regarding the tool's use in the evaluated organization. However, the results are limited to the 34 threat models available as the knowledge base. Future use of additional threat models and the inclusion of new characteristics of threat model elements may improve the quality of the generated recommendations.

Keywords: threat modeling; recommendation system; software security.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AST	Árvore Sintática Abstrata
CAPEC	*Common Attack Pattern Enumeration and Classification*
CC	Controle Contornado
CF	Controle Falho
CIA	*Confidentiality, Integrity, e Availability*
CWE	*Common Weakness Enumeration*
DFD	Diagrama de Fluxo de Dados
DPO	*Data Protection Officer*
F	Força
FC	Frequência de Contato
FP	Falso Positivo
FN	Falso Negativo
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
ID	*Identificador*
IDE	Integrated Development Environment
JSON	*JavaScript Object Notation*
LGPD	Lei Geral de Proteção de Dados
LINDDUN	*Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance*
ML	*Machine Learning* (Aprendizado de máquina)
MP	Magnitude da Perda
NR	Número de Registros
NLP	Natural Language Processor Processamento Natural de Linguagem

NTD	Número de Titulares de Dados
OWASP	Open Worldwide Application Security Project
PA	Probabilidade de Ação
PR	Período de Retenção
PV	Positivo Verdadeiro
PYTM	*Pythonic Threat Modeling*
ROC	*Rank Order Centroid*
RM	Ranking Médio
RSL	Revisão Sistemática da Literatura (RSL)
RSSE	*Recommender System for Software Engineering*
SOC	*security operation center*
STRIDE	*Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service e Elevation of Privilege*
SVD	*Singular Value Decomposition (SVD)*
SVM	*Support Vector Machines (SVM)*
TAM	*Technology Acceptance Model*
TTD	Tipo de Titular do Dado
TTF	*Task Technology Fit*
TDP	Tipo de Dado Pessoal
UTML	*Unified Threat Modeling Language*
V	Vulnerabilidade
VA	Valor do Ativo
VP	Valor da Privacidade
XML	*eXtensible Markup Language*
YAML	*YAML Ain't Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo, são apresentados a motivação para a realização da pesquisa, assim como os objetivos da pesquisa e a metodologia utilizada para sua execução.

1.1 Justificativa e Definição do Problema

Com a expansão da transformação digital dos governos e da sociedade, tem-se uma maior demanda, em curto prazo, pelo desenvolvimento e pela adoção de sistemas de informação para a execução de atividades críticas.

O desenvolvimento de software pode ser compreendido como uma lista de passos inter-relacionados, na qual o passo posterior depende do anterior, para a construção de um sistema ou software. Estes passos são comumente agrupados em um ciclo de vida, com as fases de concepção, requisitos, design, codificação, testes, implantação e manutenção (BEHERA; DASH; PAREEK, 2021).

Diversos modelos, como cascata, espiral, *agile*, entre outros, foram elaborados e podem ser utilizados para desenvolver software. Entretanto, estes modelos não consideram, por padrão, a segurança da informação como parte de suas atividades. Esta situação torna-se ainda mais relevante quando nos deparamos com o aumento da exploração de falhas de segurança nos sistemas de informação por agentes maliciosos (BeyondTrust, 2023). De acordo com o Departamento de Segurança Interna dos Estados Unidos (GASIBA et al., 2021), 90% dos incidentes de segurança são derivados de falhas de design e codificação durante a construção do software.

Em busca de tornar o processo de construção e o software mais seguros, diversas metodologias de desenvolvimento seguro foram propostas na academia, no governo e na indústria, como, por exemplo, Microsoft Software Development Life Cycle, McGraw's Secure Software Development Lifecycle Process e NIST 800-218 (KUDRIAVTSEVA; GADYATSKAYA, 2022). Estes métodos buscam reduzir a exposição do software aos agentes maliciosos através da inclusão de práticas e ferramentas de segurança durante todas as fases do ciclo de vida de desenvolvimento. Uma das práticas comumente recomendadas por estas metodologias é a modelagem de ameaças. Atualmente, ela tem sido qualificada como uma das mais importantes atividades quando se busca desenvolver software seguro (BERNSMED et al., 2022), e tem sido adotada em diversas áreas de conhecimento, como, por exemplo, em Internet of Things (CASOLA et al., 2019), dispositivos médicos (LECHNER; STRAHONJA; STAPIĆ, 2022), veículos autônomos (GHOSH et al., 2023), smart cities (TOK; CHATTOPADHYAY, 2023) e na defesa (CHO; KIM, 2025).

A modelagem de ameaças é um conjunto de técnicas que visa identificar, de forma antecipada, as ameaças ao software em desenvolvimento, tornando-o mais seguro (SHOSTACK,

2014). A sua execução, normalmente, segue a seguinte estrutura: definição do escopo, modelagem do sistema, elicitação das ameaças e revisão (YSKOUT et al., 2020). A partir da definição das ameaças identificadas, é possível selecionar os controles de segurança que devem ser aplicados ao sistema para reduzir o risco de segurança.

Existem diversos métodos e técnicas para a modelagem de ameaças, como, por exemplo, STRIDE, CLASP e OCTAVE (TUMA; CALIKLI; SCANDARIATO, 2018). Entretanto, a modelagem de ameaças ainda é considerada uma disciplina que possui baixo nível de maturidade, nos termos de pesquisa, suporte de ferramentas e abordagens práticas (YSKOUT et al., 2020).

Estes desafios ainda se somam à percepção de que a modelagem de ameaças ainda possui práticas com um alto custo manual (XIONG; LAGERSTRÖM, 2019), exige um perfil profissional altamente capacitado (TUMA et al., 2020) e é suscetível a erros (ROSA et al., 2022). Este cenário desafiador pode ser compreendido como oportunidade para a definição de meios que automatizam a modelagem de ameaças e que utilizem o conhecimento de especialistas de segurança da informação. De acordo com (YSKOUT et al., 2020), um desafio relevante para o tema é como tornar o uso de bases de conhecimento mais efetivo na prática.

Uma das formas possíveis de endereçar este problema é através do uso de sistemas de recomendação para engenharia de software (Recommender System for Software Engineering - RSSE). Um RSSE é uma aplicação que fornece dados relevantes para uma tarefa de engenharia de software em um determinado contexto (ROBILLARD et al., 2014). Seu funcionamento se dá através da sugestão dos melhores itens para atender a uma necessidade, sendo uma abordagem similar aos sistemas de recomendação utilizados na sugestão de produtos (GASPARIC; JANES, 2016). Os RSSEs têm sido utilizados para prover recomendações úteis para exemplos de código, componentes de terceiros, documentação, entre outros (ROCCO et al., 2021a). Segundo (FERREIRA; SILVA; URIARTE, 2023), no contexto da segurança da informação, os sistemas de recomendação têm sido aplicados em diversas áreas, como: predição de ataques, detecção de código inseguro, reconhecimento de *malware*, monitoramento, priorização de alertas de segurança, entre outras.

Considerando-se a lista de ameaças derivada de um design de um produto de software como uma representação da lista top-N de itens de um sistema de recomendação, pode-se considerar que, para o contexto da modelagem de ameaças, é possível utilizar um RSSE para auxiliar o trabalho do desenvolvedor na atividade da modelagem de ameaças. A automação da elicitação das ameaças por um RSSE permite que, durante o desenvolvimento de um novo software, sejam sugeridas ameaças com base nas escolhas anteriores de outros profissionais de segurança da informação. E à medida que novos modelos de ameaças são finalizados pelos profissionais, a base de conhecimento é evoluída, beneficiando, através de novas sugestões, os futuros modelos a partir deste novo conhecimento gerado.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral desta pesquisa é propor um sistema de recomendação capaz de automatizar a etapa de elicitação de um processo de modelagem de ameaças em softwares a partir do reuso de conhecimento proveniente de modelos de ameaças previamente elaborados em uma organização.

1.2.2 Objetivos Específicos

Os objetivos específicos desta pesquisa são:

- Realizar uma revisão da literatura sobre as ferramentas automatizadas para modelagem de ameaças para a identificação das técnicas mais utilizadas e detalhar o funcionamento destas ferramentas;
- Estruturar e elaborar um dataset inicial de modelos de ameaças, da hierarquia de termos utilizados e do catálogo de ameaças de uma organização para reuso pelo sistema de recomendação;
- Desenvolver a abordagem e a ferramenta para recomendação de ameaças a partir do cálculo da similaridade entre ameaças de um novo modelo e do registro histórico dos modelos de ameaças de uma organização;
- Avaliar a ferramenta com o sistema de recomendação através de uma avaliação *offline* utilizando métricas comuns a sistemas de recomendação, avaliação *online* centrada nos potenciais usuários da ferramenta na Organização X e uma avaliação online em um uso real da ferramenta na Organização X.

Com base no contexto e nos objetivos definidos, a seguinte questão de pesquisa foi estabelecida:

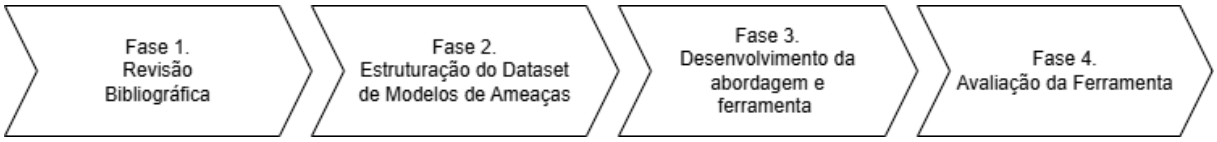
- Questão de Pesquisa: Como automatizar o processo de modelagem de ameaças em softwares utilizando o conhecimento prévio de modelos de ameaças?

1.3 Metodologia

A pesquisa é classificada quanto à natureza como aplicada, pois seu objetivo é solucionar um problema comum às organizações que trabalham com modelagem de ameaças. Quanto aos fins é uma pesquisa descritiva, pois visa descrever a modelagem de ameaças em um contexto de uma ferramenta que faça a automação da elicitação de ameaças por um sistema de recomendação

em uma organização pública federal. Quanto à forma de abordagem, é considerada quantitativa e qualitativa, de forma a aprofundar a compreensão das dificuldades no contexto do uso automatizado da modelagem de ameaças.

Figura 1 – Etapas Metodologia



Fonte: O Autor

Para alcançar os resultados esperados, adotaram-se as fases ilustradas na Figura 1, conforme descritas a seguir:

- **Revisão Bibliográfica:** A primeira fase da pesquisa consistiu na revisão bibliográfica. Inicialmente, foi utilizada uma abordagem ad hoc para compreender e identificar os fatores que dificultam a adoção da modelagem de ameaças. Em seguida, a partir da análise de uma revisão sistemática da literatura (GRANATA; RAK, 2023), identificaram-se quais as ferramentas existentes que automatizam a modelagem de ameaças, as características para avaliação de uma ferramenta de modelagem de ameaças e os métodos de elicitação automática existentes. Adicionalmente, foi realizado um detalhamento das ferramentas selecionadas para comparação. No contexto geral, foram observados os principais temas relacionados com a pesquisa, como desenvolvimento seguro, modelagem de ameaças, sistemas de recomendação, dentre outros.

- **Estruturação do dataset para o sistema de recomendação de ameaças:** Esta fase foi responsável por definir e executar o protocolo de coleta e tratamento para a construção de um dataset a ser utilizado pelo Sistema de Recomendação. O resultado foi um conjunto de modelos de ameaças, descrito em um formato estruturado, e um vocabulário de termos referentes aos elementos que compõem um modelo de ameaças.

- **Desenvolvimento da abordagem e da ferramenta para recomendação de ameaças:** Esta fase foi responsável pela definição dos parâmetros e do método de funcionamento do sistema de recomendação proposto. Foram definidas algumas características importantes para o sistema, como: a unidade de comparação para recomendação das ameaças; as funções de similaridade para a comparação; a função de recomendação e seus pesos adotados via consenso de especialistas; os componentes e atividades envolvidos no uso da ferramenta; a arquitetura utilizada pela ferramenta; e detalhes de implementação.

- **Avaliação da ferramenta:** Esta etapa teve o objetivo de avaliar a abordagem e a ferramenta apresentada. Optou-se por utilizar avaliações *offline* e *online* do sistema de recomendação da ferramenta. Na avaliação *online* foi realizada inicialmente uma avaliação centrada no

usuário, para avaliar a aceitação e o ajuste tarefa-tecnologia fundamentados no modelo *technology acceptance model* (TAM) (DISHAW; STRONG, 1999) integrado ao modelo *task-technology fit* (TTF) (GOODHUE; THOMPSON, 1995). Foram utilizadas entrevistas semiestruturadas (perguntas quantitativas e qualitativas) com os participantes do processo de utilização da ferramenta. Por fim, foi realizada uma avaliação de uso real da ferramenta em um produto de software da Organização X avaliada.

1.4 Aplicabilidade

A presente pesquisa propõe o desenvolvimento do sistema *Threat Copilot*, um sistema de recomendação para engenharia de software (RSSE), fundamentado em algoritmos de similaridade entre ameaças.

A aplicabilidade deste estudo reside em facilitar a elicitação de ameaças durante a modelagem de ameaças, oferecendo aos profissionais da área de segurança da informação e desenvolvimento de software um recurso capaz de otimizar a identificação de ameaças de software e a definição de controles de segurança relacionados.

Dessa forma, a proposta busca contribuir tanto para o avanço acadêmico na definição de sistemas de recomendação no contexto da engenharia de software quanto para a adoção prática na Organização X avaliada nesta pesquisa e ainda em organizações com contexto semelhante ao avaliado que buscam fortalecer suas estratégias de desenvolvimento seguro.

1.5 Estrutura do Documento

Esta dissertação está organizada em seis capítulos, conforme descrito a seguir:

- **Capítulo 2 – Fundamentação Teórica:** Revisa os conceitos fundamentais sobre desenvolvimento seguro, modelagem de ameaças, sistemas de recomendação, incluindo suas definições, classificações e critérios de seleção, finalizando com a análise de trabalhos relacionados e detalhamento das ferramentas. Contribuí para dar embasamento teórico ao tema deste trabalho.
- **Capítulo 3 – O Sistema de Recomendação:** Detalha o Sistema de Recomendação proposto, contemplando os aspectos relacionados à construção do dataset, a heurística utilizada para a seleção das unidades de ameaça, as atividades e componentes da arquitetura da ferramenta, assim como detalhes da sua implementação.
- **Capítulo 4 – Avaliação do Sistema de Recomendação:** Apresenta os resultados obtidos a partir da avaliação *offline* de métricas comuns a sistemas de recomendação. Também são apresentados os resultados obtidos a partir de avaliações *online* centradas no usuário e no uso real da ferramenta em um projeto dentro da organização avaliada.

- **Capítulo 5 – Discussão das Avaliações:** Analisa os resultados à luz da literatura revisada, interpretando os dados obtidos no contexto da organização pública de tecnologia da informação avaliada.
- **Capítulo 6 – Conclusão:** Sintetiza os principais achados da pesquisa, apresenta suas contribuições teóricas, metodológicas e práticas, discute as limitações do estudo, propõe direções para pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve os principais conceitos utilizados nesta pesquisa. Inicialmente é apresentada uma introdução ao desenvolvimento seguro e à modelagem de ameaças. Logo após, são apresentados os conceitos de sistemas de recomendação e sua aplicação na engenharia de software. Por fim, são apresentados os trabalhos relacionados a esta pesquisa e o detalhamento das ferramentas de modelagem de ameaças automatizadas selecionadas.

2.1 Desenvolvimento Seguro

Em um mundo cada vez mais conectado e gerenciado por software, é natural o aumento do interesse de agentes maliciosos em subverter a segurança de sistemas para obter vantagens ilícitas. Atualmente, são realizadas transações bancárias, declarações de imposto de renda, cursos de pós-graduação a distância, bem como a comprovação e o recebimento de benefícios sociais, como aposentadorias, auxílios emergenciais e seguro-desemprego, por meio de aplicativos desenvolvidos pela indústria, pelo governo e pela academia. A proteção da integridade, confidencialidade e autenticidade das informações processadas por esses aplicativos constitui responsabilidade da área de cibersegurança. Segundo Solms e Niekerk (2013), a cibersegurança pode ser definida como:

> A coleção de ferramentas, políticas, conceitos de segurança, contramedidas de segurança, guias, abordagens de gerenciamento de riscos, ações, treinamentos, melhores práticas, garantia e tecnologias que podem ser usadas para proteger o ambiente cibernético e os ativos dos usuários e da organização. Estes ativos incluem dispositivos computacionais, pessoas, infraestrutura, aplicações, serviços, sistemas de telecomunicação e toda a informação transmitida e armazenada no ambiente cibernético. (SOLMS; NIEKERK, 2013)

Uma das atividades importantes da cibersegurança é a proteção das aplicações de software. Existem diversos processos para o desenvolvimento seguro de software considerando diferentes atividades e fases do desenvolvimento (KUDRIAVTSEVA; GADYATSKAYA, 2022). Para avaliar a maturidade de uma organização quanto ao seu processo de desenvolvimento seguro, foi criado um modelo de maturidade em segurança de software (*Build Security In Maturity Model*) (Synopsys, 2022). Ele define um conjunto de 125 atividades mais comuns para a segurança no desenvolvimento de software, classificadas em 4 domínios e 12 práticas. A Figura 2 abaixo exibe os domínios e práticas mais comuns em 254 organizações analisadas.

Dentre as atividades, a modelagem de ameaças é considerada a mais popular para design seguro (AL-MATOUQ et al., 2020) e a mais importante (MCGRAW, 2006) em um processo de desenvolvimento. Ela se enquadra no domínio de SDL Touchpoints e na prática de “Análise de Arquitetura”.

Figura 2 – Domínios e Práticas de Desenvolvimento Seguro

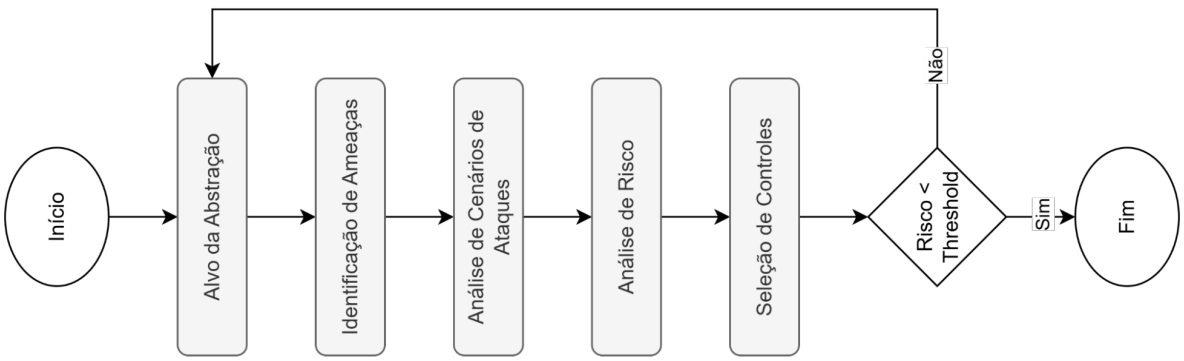
Fonte: (Synopsys, 2022)

Existem diversas definições para modelagem de ameaças (XIONG; LAGERSTRÖM, 2019). Entretanto, para este trabalho, será utilizada a definição proposta por Cho e Kim (2025), por ser a mais recente e integrar as várias concepções presentes em padrões internacionais da indústria, como ISO/IEC 27005:2022 e NIST SP 800-154. Além disso, essa definição torna explícitas as fases que compõem o processo de modelagem de ameaças. Segundo os autores, a modelagem de ameaças é definida como:

> Uma técnica de análise de segurança que abstrai o sistema-alvo em um modelo, identifica sistematicamente as ameaças potenciais, as prioriza por meio de uma análise de risco dos cenários de ataque e elabora contramedidas apropriadas. (CHO; KIM, 2025)

A partir dessa definição, Cho e Kim (2025) descrevem as fases mais comuns de um processo de modelagem de ameaças. A Figura 3 define a sequência do processo nas fases: alvo da abstração, identificação das ameaças, análise de cenários de ataque, análise de risco e seleção de controles.

Figura 3 – Processo de Modelagem de Ameaças



Fonte: (CHO; KIM, 2025)

A primeira fase consiste na remoção de todas as informações não relacionadas à segurança da informação sob o alvo da modelagem, concentrando-se exclusivamente nos dados relevantes à

segurança para a identificação de ameaças potenciais. Esse modelo abstraído também contempla a identificação dos ativos que necessitam de proteção, tais como hardware, software, firmware, dados, pessoas e recursos financeiros.

Na segunda fase, as ameaças potenciais são identificadas a partir do modelo abstraído, utilizando todos os recursos disponíveis.

A terceira fase está relacionada à análise de cenários de ataque e tem como objetivo avaliar se as ameaças identificadas podem ser exploradas de forma realista. Essa etapa confirma a aplicabilidade das ameaças em um contexto de ataque do mundo real.

A quarta fase envolve a análise e avaliação de riscos. Uma vez estabelecidos os cenários de ataque, o nível de risco de cada cenário é avaliado e priorizado.

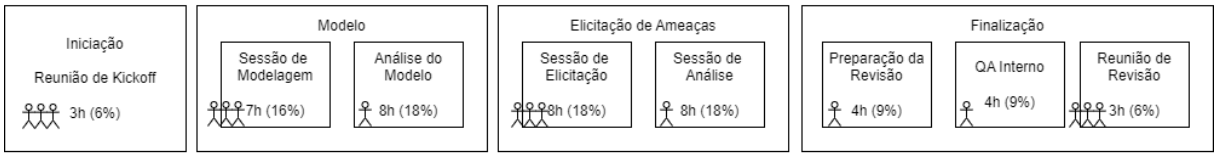
A quinta fase compreende a seleção de contramedidas destinadas a mitigar os cenários de ataque identificados, de acordo com as prioridades previamente estabelecidas.

O processo de modelagem de ameaças é iterativo, reiniciando-se na etapa de abstração, a fim de assegurar a adequação das contramedidas selecionadas. Esse ciclo prossegue até que não sejam identificados novos cenários exploráveis ou todos os níveis de risco sejam considerados aceitáveis. Mudanças na unidade de análise podem exigir novas análises, uma vez que alterações na abstração podem modificar a lista de ameaças.

Do ponto de vista da engenharia de software, o objetivo da modelagem de ameaças é identificar as fraquezas antes que se tornem parte do sistema através da codificação ou implantação, desta forma, sendo possível realizar ações corretivas o mais cedo possível.

Yskout et al. (2020) define que o processo de modelagem de ameaças em uma organização segue normalmente os seguintes passos: reunião inicial de preparação, definição do modelo, elicitação das ameaças e revisão final da modelagem de ameaças. A Figura 4 abaixo apresenta as atividades da modelagem de ameaça no contexto de uma organização de desenvolvimento de software, considerando a participação de um ou mais indivíduos envolvidos em cada atividade e o tempo esperado.

Figura 4 – Atividades de um processo de modelagem de ameaças em uma organização



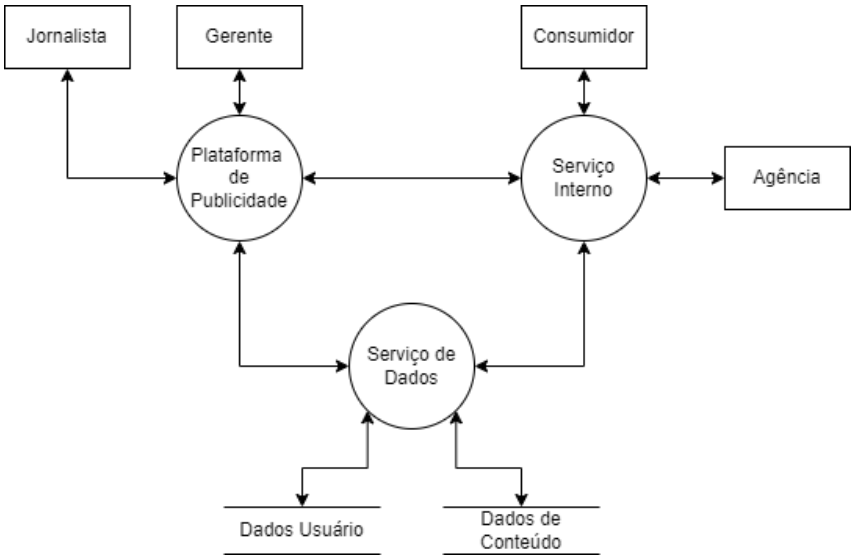
Fonte: (YSKOUT et al., 2020)

Na reunião inicial (kick-off), os participantes estabelecem de forma conjunta o escopo e o objetivo do processo de modelagem de ameaças. Após esta definição, é realizada uma sessão de modelagem onde o sistema, seus ativos e relacionamentos são detalhados e revisados por um especialista em modelagem de ameaças. De posse do modelo do sistema, as ameaças são

identificadas, analisadas e ranqueadas por um especialista em segurança, que, ao final de todo o processo, prepara uma reunião final para revisão do modelo e divulgação das ameaças. Caso o modelo não alcance a qualidade desejada, um novo ciclo de modelagem pode ser realizado.

A modelagem de ameaças pode ser realizada através de diversos processos existentes tanto na indústria como na academia (XIONG; LAGERSTRÖM, 2019). Dentre eles, o mais conhecido é o Microsoft STRIDE (SCANDARIATO; WUYTS; JOOSEN, 2015). Ele consiste em guiar o analista de segurança por diversas fases do processo de modelagem de ameaças. Na fase de modelo, é necessário descrever e detalhar o escopo do sistema analisado por meio de um diagrama de fluxo de dados (YOURDON, 1989). Este diagrama representa de forma padronizada os elementos do sistema, utilizando retângulos para os atores, círculos para os processos, linhas tracejadas para as fronteiras de confiança, duplas linhas horizontais para os armazenamentos de dados e setas direcionais para os fluxos de informação que estabelecem os relacionamentos entre os componentes. A Figura 5 abaixo apresenta um exemplo de diagrama de dados de um sistema de publicidade.

Figura 5 – Exemplo de diagrama de fluxo de dados para modelagem de ameaças



Fonte: Adaptado de (SCANDARIATO; WUYTS; JOOSEN, 2015)

Voltando à Figura 5, observam-se os atores Jornalista, Gerente, Consumidor e Agência, que interagem com os processos internos do sistema. Entre os processos, destacam-se a Plataforma de Publicidade, o Serviço Interno e o Serviço de Dados, que constituem o núcleo operacional do sistema. O Serviço de Dados está conectado a dois armazenamentos de dados denominados Dados do Usuário e Dados de Conteúdo. As setas direcionais indicam o fluxo de informações entre os componentes, evidenciando o intercâmbio de dados entre os processos e os atores externos.

Na fase de elicitação, cada elemento do diagrama de fluxo de dados, seja ator, processo, armazenamento ou fluxo de dados, é analisado e mapeado, de acordo com a técnica aplicada, seja

por interação (focando nas trocas entre elementos) ou por elemento (avaliando cada componente individualmente), em uma ou mais das seis categorias de ameaças do acrônimo STRIDE:

- (S) *Spoofing*. Referente à capacidade de um atacante assumir ilegalmente a identidade de um terceiro legítimo.
- (T) *Tampering*. Referente a um atacante realizar ilegalmente a alteração de um dado íntegro, por exemplo, dados em memória ou dados em banco de dados.
- (R) Repudiation. Capacidade de um atacante negar a autoria da realização de uma ação em um sistema.
- (I) *Information Disclosure*. Referente a um atacante obter uma informação privada ilegalmente a partir do sistema. Seja pela leitura de um arquivo, pela interceptação de dados em trânsito ou ausência de controle de acesso por uma API.
- (D) *Denial of Service*. Referente à capacidade do atacante de tornar um recurso do sistema indisponível para seus usuários.
- (E) *Elevation of Privilege*. Referente à capacidade do atacante de obter de forma ilegal o acesso privilegiado a recursos normalmente protegidos.

Continuando na fase de elicitação, para cada ameaça genérica associada, é necessário identificar as ameaças concretas que devem ser consideradas através do preenchimento dos critérios de um checklist de uma biblioteca de ameaças.

Ao final deste processo, na fase de finalização, a documentação das ameaças é realizada e as contramedidas são determinadas para que sejam apresentadas ao time de desenvolvimento.

Embora a modelagem de ameaças seja um processo com poucas fases, ele consome tempo, exige a participação de profissionais especialistas e o resultado depende da experiência dos participantes. Yskout et al. (2020) identificou os critérios essenciais para adoção com sucesso de uma abordagem de modelagem de ameaças:

- **Baseado em modelos:** O uso de modelos facilita a comunicação e o entendimento do sistema por parte dos participantes da modelagem.
- **Rastreável:** Os modelos de ameaças precisam ser detalhados considerando informações relevantes e subentendidas que, quando ausentes, impactam as futuras versões do modelo.
- **Sistemático:** A modelagem de ameaças deve ser capaz de garantir que os modelos sejam completos e corretos, especialmente, quando o responsável pela modelagem possui pouca experiência. A automação e o suporte de ferramentas são importantes para garantir o apoio com o conhecimento de um especialista na identificação das ameaças.

- **Integração com o negócio:** O resultado da modelagem de ameaças deve ser integrado aos sistemas de gerenciamento de segurança da informação da organização. De forma que as ameaças e riscos identificados e classificados sejam gerenciados pelos *stakeholders* da organização.
- **Suporte a contexto:** A modelagem de ameaças deve considerar a organização em que está inserida, permitindo que modelos individuais sejam facilmente integrados ao contexto organizacional, incluindo ativos de proteção, padrões e políticas organizacionais.
- **Escalável:** A modelagem de ameaças deve ser escalável em termos de recursos que precisam ser investidos para a sua execução diante do tamanho e complexidade do sistema que será analisado.

Estes critérios são utilizados na análise dos resultados deste trabalho, avaliando cada critério com a abordagem apresentada nesta pesquisa e comparando com as percepções obtidas a partir da validação centrada no usuário.

2.2 Sistemas de Recomendação para Engenharia de Software

Os sistemas de recomendação desempenham um papel cada vez mais importante em várias áreas, desde o comércio eletrônico até o streaming de mídia, ajudando a melhorar a experiência do usuário e a fornecer sugestões personalizadas. Neste cenário, os sistemas de recomendação se tornaram indispensáveis para facilitar a navegação dos usuários e encontrar os produtos, serviços ou conteúdos que melhor atendam às suas necessidades e preferências.

Por trás do desenvolvimento dos sistemas de recomendação está a necessidade de aprimorar a experiência do usuário, economizar tempo, aumentar a satisfação geral e principalmente, a quantidade de vendas de produtos (AGGARWAL, 2016). Com a grande quantidade de dados disponíveis, a necessidade de algoritmos inteligentes capazes de filtrar e personalizar informações é crucial.

Segundo (AGGARWAL, 2016) existem diversas variantes para a formulação do problema da recomendação de itens, e os dois modelos primários são: a versão preditiva e a versão de ranqueamento.

Na versão preditiva espera-se descobrir o valor de uma avaliação para uma combinação usuário-item, por exemplo, qual a quantidade de estrelas que um usuário irá escolher para um determinado filme. Assume-se que existam dados para treinamento disponíveis, indicando a avaliação dos usuários para os itens. Com base nestes dados observados, espera-se predizer os valores não observados que serão utilizados para recomendar os filmes. Por exemplo, prever que o usuário daria quatro estrelas a um determinado filme.

Por outro lado, na versão de ranqueamento, o objetivo principal não é estimar uma avaliação exata, mas sim ordenar os itens de acordo com sua relevância para o usuário. Nesse

contexto, busca-se determinar quais são os top-N itens mais relevantes para um determinado usuário. Assim, o foco está na qualidade da ordenação gerada, garantindo que os itens mais prováveis de interesse apareçam nas primeiras posições da lista. Para o contexto desta pesquisa, a versão de ranqueamento é o modelo que será utilizado para determinar a elicitação na modelagem de ameaças.

Os sistemas de recomendação podem ser classificados de acordo com a estratégia utilizada para identificar itens potencialmente relevantes para um usuário. Essa classificação é importante, pois cada abordagem emprega diferentes fontes de informação e distintos mecanismos para inferir preferências e produzir recomendações. De acordo com Aggarwal (2016) os tipos básicos de sistemas de recomendação são de filtragem colaborativa, baseados em conteúdo e baseados em conhecimento. Eles são brevemente descritos nos tópicos abaixo:

- **Filtragem colaborativa baseada em usuários:** nesse método, as recomendações são geradas a partir das avaliações feitas por usuários Z que possuem preferências semelhantes às do usuário-alvo A. O princípio fundamental consiste em identificar quais usuários apresentam comportamento similar ao usuário A e, com base nisso, estimar o interesse do usuário-alvo em itens X que ele ainda não avaliou. Essa estimativa é geralmente obtida a partir da média ponderada das avaliações atribuídas a esses itens X pelos usuários Z similares.

- **Filtragem colaborativa baseada em item:** nesse método, o objetivo é prever a avaliação que o usuário A daria a um item-alvo X. Para isso, o sistema identifica primeiro quais itens Y são mais semelhantes ao item-alvo X. A similaridade entre itens é determinada pela comunidade, em que, de maneira geral, usuários tendem a avaliá-los de forma semelhante. Em seguida, utiliza as avaliações que o próprio usuário A fez desses itens semelhantes para estimar se ele provavelmente gostará do item X.

- **Baseado em conteúdo:** nesse método, as recomendações são geradas com base nas características dos itens e nas preferências demonstradas pelo próprio usuário. Assim, quando não há avaliações de outros usuários disponíveis, o sistema identifica itens com atributos semelhantes àqueles que o usuário avaliou positivamente, sugerindo conteúdos com perfis similares aos de seu histórico de interesse.

- **Baseado em conhecimento:** nesse método, as recomendações são produzidas a partir do relacionamento entre as características dos itens e os requisitos explicitamente informados pelo usuário. A entrada para a determinação das recomendações é um conjunto de características que o usuário deseja e as recomendações dos itens são feitas a partir do ranqueamento da similaridade dos itens com as características solicitadas.

Ricci, Rokach e Shapira (2010), Aggarwal (2016) também mencionam outros tipos de sistemas de recomendação, como os baseados em contexto, sensíveis ao tempo, baseados em loca-

lização, sociais e demográficos. Optou-se por não aprofundar a descrição destes tipos de sistemas de recomendação, uma vez que o objetivo foi apresentar apenas os modelos fundamentais.

Robillard et al. (2014) indica que os sistemas de recomendação não são restritos ao contexto de comércio eletrônico, mas também podem ser utilizados em domínios técnicos para ajudar na descoberta de informações que podem ser úteis para os trabalhadores na execução de suas tarefas. Para a engenharia de software, um sistema de recomendação é uma aplicação de software que provê itens de informação valiosos para uma atividade de desenvolvimento de software em um determinado contexto (ROBILLARD et al., 2014). Diversos sistemas de recomendação para engenharia de software foram desenvolvidos, como por exemplo, para selecionar exemplos de código, componentes de terceiros e documentação (ROCCO et al., 2021a) Como também, para a recomendação de casos de teste para sistemas (FILHO, 2021).

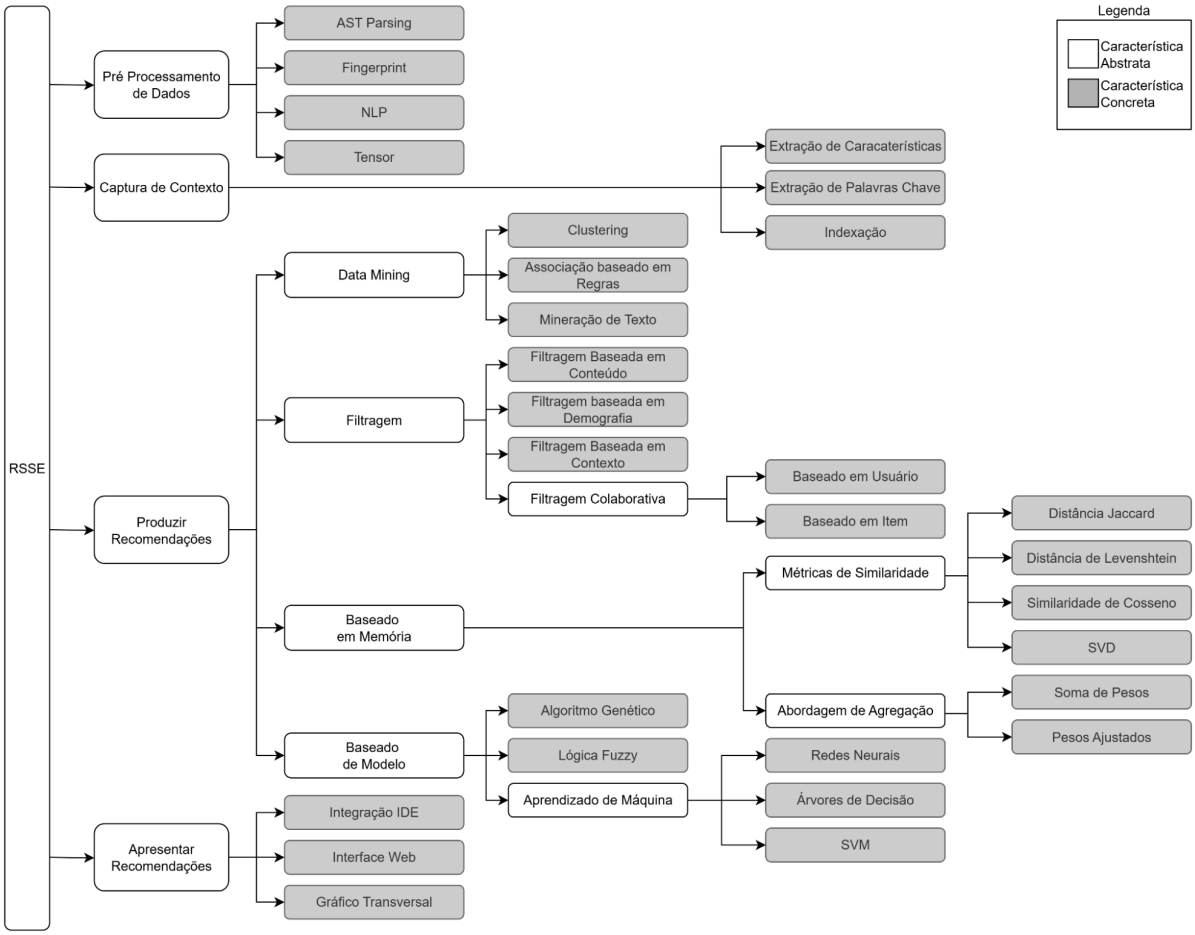
Segundo Rocco et al. (2021b) é importante classificar os sistemas de recomendação para facilitar o entendimento das técnicas e ferramentas utilizadas durante o seu desenvolvimento e uso. As principais funcionalidades de design de um sistema de recomendação são: pré-processamento de dados, captura do contexto, produzir recomendações (data mining, filtragem, baseado em memória e baseado em modelo) e, por fim, apresentar recomendações. A Figura 6 apresenta as principais funcionalidades de design de um sistema de recomendação.

Na fase de pré-processamento dos dados, diversas técnicas e ferramentas são utilizadas para extrair informações relevantes de diferentes fontes de dados. Sejam dados estruturados, como, por exemplo, código-fonte ou documentos XML, ou dados não estruturados, como documentos, posts de blogs, etc. Para isso são comumente utilizadas técnicas de análise sintática e semântica, como *parsing* de árvores sintáticas abstratas (AST) de código-fonte, *fingerprint* que mapeia uma string em uma sequência de bits, processamento natural de linguagem (NLP) e *tensors* que codificam a partir de dados não estruturados as relações entre dados e preferência dos usuários.

Na fase de captura de contexto, o contexto é extraído do ambiente de uso para habilitar o mecanismo de recomendação. Uma técnica bem comum, amplamente utilizada no domínio de aprendizado de máquina (ML), é a extração de características *(Feature Extraction)*. Outra técnica, utilizada bastante no contexto de avaliação de código-fonte, é a extração de palavras-chave. Por fim, a indexação é uma técnica comum dos engenhos de busca para recuperar rapidamente informações relevantes.

Na fase de produção de recomendação, os algoritmos de recomendação são escolhidos e executados para produzir sugestões relevantes ao contexto do usuário. Estão divididos em quatro grupos principais de abordagens. A primeira, denominada *Data Mining*, abrange técnicas como clustering, associação baseada em regras e mineração em texto, voltadas à extração de padrões e correlações entre elementos do conjunto de dados. A segunda, Filtragem, compreende métodos de filtragem que incluem baseados em conteúdo, demográficos, baseados em contexto e colaborativos (nas variações baseadas em usuário ou item). O terceiro grupo de abordagem,

Figura 6 – Principais Funcionalidades do Design de um Sistema de Recomendação



Fonte: (ROCCO et al., 2021a)

baseado em memória, pelo uso de dados históricos por meio de medidas de similaridade, como, por exemplo, jaccard, levenshtein e cosseno, e técnicas de agregação, como *Singular Value Decomposition (SVD)*, soma de pesos e pesos ajustados. Por fim, a abordagem baseada em modelos emprega modelos de aprendizado de máquina, destacando-se as redes neurais, árvores de decisão, *Support Vector Machines (SVM)*, algoritmos genéticos e lógica *fuzzy*, que buscam generalizar o comportamento dos dados e aprimorar a precisão das recomendações.

Na última fase, os itens de recomendação gerados precisam ser apresentados adequadamente ao desenvolvedor. Para isso, diversas estratégias podem envolver tecnologias diferentes, incluindo o desenvolvimento de extensões para IDEs, de interfaces dedicadas baseadas na Web ou de grafos transversais que permitem a navegação entre os itens recomendados.

Esta dissertação entende a modelagem de ameaças como parte de uma das atividades presentes em um processo de desenvolvimento seguro de software para engenharia de software, sendo uma das atividades que podem se beneficiar de um sistema de recomendação para elicitar ameaças com base em modelos construídos a partir de diagramas de fluxo de dados.

2.3 Trabalhos Relacionados

Recentemente, Granata e Rak (2023) realizaram uma Revisão Sistemática da Literatura (RSL) utilizando a metodologia proposta por Kitchenham et al. (2009), com o objetivo de responder a seguintes questões de pesquisa:

- Q1: Quais metodologias são usadas para produzir modelos de ameaças automaticamente?
- Q2: Quais ferramentas de código aberto são usadas para apoiar as metodologias de modelagem de ameaças automáticas?

O resultado da RSL resultou em cinquenta estudos primários que propuseram uma abordagem de automação/semi-automação do processo de modelagem de ameaças e nove estudos apresentaram nove ferramentas de terceiros apoiando a modelagem de ameaças e uma desenvolvida pelos autores. Sendo as ferramentas:

- TAM tool (SCHAAD; BOROZDIN, 2012);
- AutSEC (FRYDMAN et al., 2014);
- STS-tool (MELAND et al., 2014);
- MetaGME (MARTINS et al., 2015);
- Microsoft Threat Modeling Tool (Microsoft, 2018);
- OWASP Threat Dragon (OWASP, 2020);
- SPARTA (SION et al., 2018);
- TAMELESS (MELAND et al., 2014);
- CoreTM (ASSEN et al., 2022);
- e Sla-generator (GRANATA; RAK; SALZILLO, 2022).

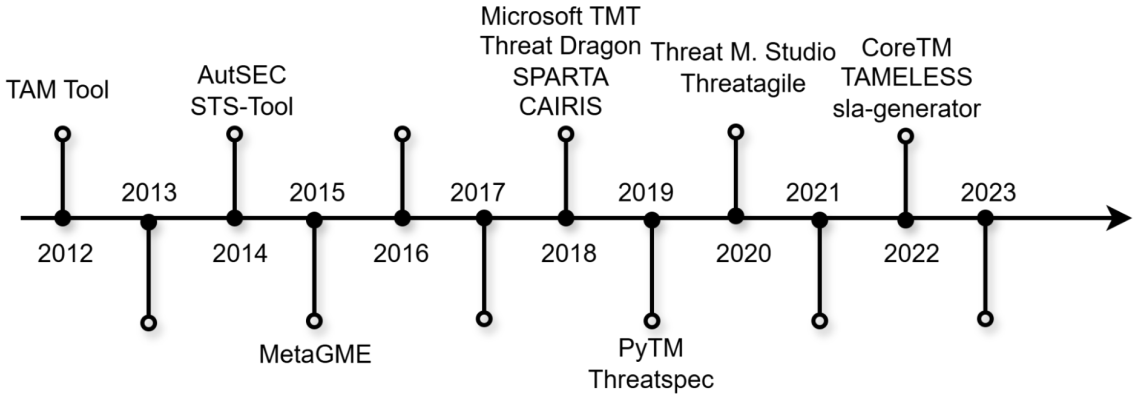
Adicionalmente, na mesma RSL, com objetivo de enriquecer e ter uma lista mais completa, os autores incluíram, através de uma consulta específica aos buscadores, com parâmetros não informados, mais cinco novas ferramentas:

- CAIRIS (FAILY, 2018);
- Threats Manager Studio (TMS, 2020);
- ThreatSpec (ThreatSpec, 2019);
- PyTM (OWASP, 2019c);

- e Threagile (Threatagile, 2020).

Granata e Rak (2023) ressaltaram que o resultado final da lista de ferramentas selecionadas derivou dos resultados da RSL e da inclusão manual. O critério deste acréscimo manual não foi claramente definido pelos autores. O foco observado pelos autores é as técnicas de modelagem de ameaças nas quais cada ferramenta se baseia, sendo a ferramenta necessária "apenas" para validá-las. A Figura 7 apresenta a linha do tempo das ferramentas com métodos de automação na elicitação de ameaças.

Figura 7 – Linha do tempo da publicação das ferramentas de modelagem de ameaças.



Fonte: O Autor.

As ferramentas de modelagem automatizada de ameaças tem seu início em 2012 com a ferramenta TAM Tool e suas últimas ferramentas são de 2022. Considerando o escopo desta pesquisa, decidiu-se selecionar as ferramentas identificadas por Granata e Rak (2023) a partir dos seguintes critérios:a) ferramentas que trabalham os modelos de ameaças a partir de diagramas de fluxo de dados, b) permitam a elicitação automática de ameaças, c) possibilitem a modelagem gráfica, seja pela criação manual do usuário ou pela geração a partir de código e d) não reutilizem o método de elicitação automática de ameaças.

Baseado nestes critérios, foram eliminadas as seguintes ferramentas:

- TAM Tool: Seu método de elicitação de ameaças baseia-se na abordagem STRIDE, já contemplada pela ferramenta Microsoft Threat Modeling Tool.
- MetaGME: Aplica uma extensão voltada ao contexto de sistemas ciberfísicos, também utilizando a ferramenta Microsoft Threat Modeling Tool já selecionada.
- STS-Tool: Não utiliza diagramas de fluxo de dados, mas sim uma linguagem própria para representação dos componentes e de seus relacionamentos (STS Modeling Language).

- ThreatSpec: Não utiliza diagramas de fluxo de dados, e sim anotações inseridas nos comentários do código-fonte para gerar o modelo de ameaças. A ferramenta requer, portanto, a existência do código-fonte do software para a realização da modelagem.
- CORETM: A automação está relacionada apenas à geração de relatórios de ameaças, sendo a elicitação de ameaças realizada de forma manual.
- CAIRIS: A automação está restrita à geração de diagramas de fluxo de dados, não abrangendo a etapa de elicitação de ameaças.
- TAMELESS: Não utiliza diagramas de fluxo de dados, mas sim a linguagem Prolog para descrever modelos de ameaça aplicados a sistemas ciberfísicos.

Por outro lado, as oito ferramentas restantes foram selecionadas para um detalhamento do seu funcionamento. A RSL de Granata e Rak (2023) apresentou uma análise detalhada das ferramentas *Microsoft Threat Modeling Tool, OWASP Threat Dragon, PyTM e sla-generator*. Complementarmente, como parte da revisão bibliográfica desta dissertação, foi realizado o detalhamento das ferramentas *AutSec, SPARTA, Threat Manager Studio e Threat-agile*.

As próximas subseções descrevem estas ferramentas e suas técnicas para elicitação de ameaças de forma automatizada.

2.3.1 Ferramenta AutSec

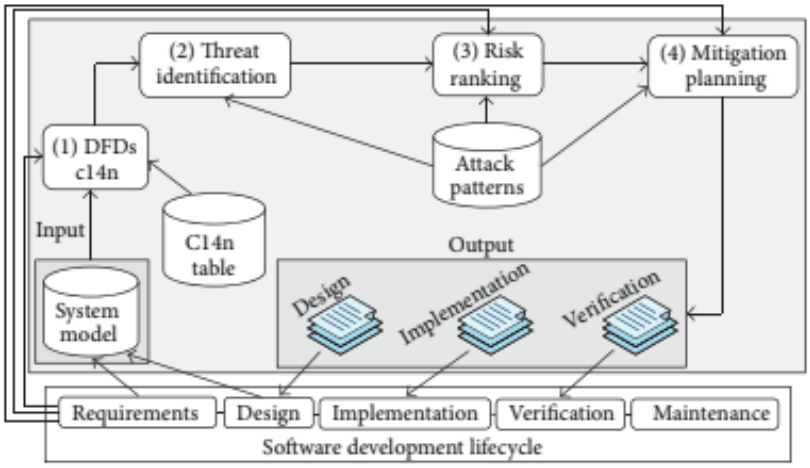
O propósito da ferramenta AutSec (FRYDMAN et al., 2014) avaliada é aplicar o *security by design*, assegurando que durante o desenvolvimento de software as ameaças identificadas sejam tratadas mais cedo e automatizar todas as operações do processo de modelagem de ameaças para permitir que engenheiros não especialistas em segurança da informação possam desenvolver software seguro.

O processo de elicitação automática da ferramenta e emissão dos relatórios de ameaças e mitigação é feito a partir dos passos definidos na Figura 8.

A ferramenta tem como entrada um DFD *(System model)*, elaborado na fase de requisitos do projeto, contendo os processos, fluxos, armazéns de dados, entidades externas e fronteiras de confiança.

1. **Canonização dos elementos do DFD:** Normalização dos rótulos definidos pelos desenvolvedores. Nesta fase, os termos concretos são mapeados em termos (*labels*) mais genéricos, armazenados na C14ntable. Por exemplo, o termo Apache, que normalmente se refere ao *Apache2 Web Server*, é convertido em *Web Server*. Quando o mapeamento não é identificado, perguntas são feitas ao desenvolvedor para a realização do mapeamento.
2. **Identificação das Ameaças:** Cada ameaça é definida na forma de árvores de ameaças, que são subgrafos que devem ser identificados no DFD. Cada ameaça é representada por

Figura 8 – Arquitetura da ferramenta AutSec.



Fonte: (FRYDMAN et al., 2014)

uma árvore de identificação (*identification tree*), que descreve as condições estruturais e lógicas necessárias para que aquela ameaça possa ocorrer dentro do modelo de fluxo de dados (DFD). Essas árvores funcionam como subgrafos que precisam ser encontrados no DFD. Cada nó da árvore representa um elemento do sistema (como processos, fluxos de dados, entidades externas ou armazenamentos), e cada aresta representa uma relação entre esses elementos. A correspondência entre a árvore de ameaça e o DFD é verificada automaticamente: a) Se todos os nós e condições lógicas da árvore são satisfeitos, por exemplo, um fluxo HTTP atravessando uma fronteira de confiança sem uso de criptografia, a ameaça é incluída na lista de ameaças; b) Se algum requisito não é atendido, por exemplo, um canal seguro HTTPS, a ameaça não é incluída.

3. **Ranqueamento dos Riscos e Eliminação das Ameaças:** Nesta etapa, o risco de cada ameaça é calculado utilizando a definição de Probabilidade x Impacto definida pela base de conhecimento CAPEC. O resultado é um valor numérico que reflete a severidade do risco. Para facilitar a priorização, o analista define um limiar mínimo de relevância, chamado de *threshold*. Esse limiar representa o ponto a partir do qual uma ameaça é considerada para o tratamento via mitigação. O objetivo desta etapa é priorizar as ameaças com maior risco para o sistema, naturalmente, otimizando os custos envolvidos na implementação de controles de segurança.

4. **Planejamento de Mitigação:** Nesta última fase, a ferramenta gera relatórios sob a perspectiva de design, implementação e testes. O relatório indica os controles de segurança que podem ser adotados para mitigar a ameaça identificada. Esta definição é feita a partir de um novo conceito definido como árvore de mitigação. Na árvore são definidos: objetivo principal, as ações, requisitos técnicos, os controles concretos, o custo estimado de implementação do controle, definido por *experts*, e dependência de outras mitigações.

Nela são avaliados o somatório dos custos e as alternativas existentes para atender a cada controle. Ao final, o controle possível de menor custo é selecionado e apresentado, com o detalhamento da implementação e dos testes necessários para a verificação do sucesso da mitigação.

Ao final, os três relatórios de design, implementação e testes agregam as informações da base de conhecimento definida na Tabela 1.

Tabela 1 – Base de Conhecimento de Ameaças da *AutSec*

Campo	**Descrição**
Título	Título da ameaça.
Componentes Afetados	Listas dos caminhos que relacionam o agente da ameaça ao ativo afetado.
Descrição	Regras booleanas que determinam a exclusão da ameaça.
Exemplo	Exemplo prático ilustrando uma vulnerabilidade associada à ameaça.
Referências	Links para bases de conhecimentos externas como CAPEC, CWE e OWASP.
Especificação de controles de segurança	Detalhamento de como aplicar os controles de segurança no contexto da ameaça.

Fonte: Autor.

A Tabela 1 apresenta a estrutura que organiza as informações necessárias para a identificação e análise de ameaças. Cada registro inclui o título da ameaça, os componentes afetados e uma descrição baseada em regras booleanas que definem sua aplicabilidade. Além disso, são fornecidos exemplos práticos que ilustram vulnerabilidades associadas, referências a bases externas como CAPEC, CWE e OWASP e a especificação de controles de segurança, que detalham as medidas recomendadas para mitigação.

2.3.2 Ferramenta Microsoft Threat Modeling Tool

A Microsoft TMT é uma ferramenta gratuita de modelagem de ameaças, sendo o resultado de anos de desenvolvimento (SHI et al., 2021). De acordo com Granata e Rak (2023) ela é a ferramenta mais utilizada na definição dos métodos de elicitação automática de ameaças. Cada modelo é construído com base em um padrão (*template*) que define os tipos de componentes e suas especializações do diagrama, como, por exemplo, um fluxo de dados genérico pode ser definido por um tipo mais especializado, como *http, https e smb*. Este mesmo padrão também determina as regras responsáveis por determinar a elicitação automática de ameaças.

A ferramenta considera que cada elemento de um diagrama de fluxo de dados é caracterizado por um par de etiquetas que é definido pelo tipo e valor do componente. Os tipos e os valores são prefixados pelo template. Por exemplo, para um banco de dados Redis armazenado em um provedor de nuvem da Azure, o tipo de componente seria *Generic Data Store* e o valor seria *Azure Redis Cache*.

Seu modelo de elicitação automática baseia-se em relacionamento. Desta forma, considera-se o tipo de interação entre os dois componentes e a verificação das propriedades destes componentes. A elicitação ocorre diante de um modelo completo de ameaças definido; a ferramenta analisa cada interação entre os elementos do modelo e seleciona as ameaças da base de conhecimentos da ferramenta com base em uma consulta considerando os filtros de inclusão e de exclusão de cada ameaça presente na base de conhecimento. Esta consulta considera o tipo do componente, o valor especializado do componente e as propriedades envolvidas. Os campos que compõem a base de conhecimento são apresentados na Tabela 2.

Tabela 2 – Base de Conhecimento de Ameaças da *Microsoft Threat Modeling Tool*

Campo	**Descrição**
ID	Identificação da ameaça.
Filtro de Inclusão	Regras booleanas que determinam a inclusão da ameaça.
Filtro de Exclusão	Regras booleanas que determinam a exclusão da ameaça.
Título Curto	Descrição curta do título da ameaça.
Categoria	Classificação da ameaça no acrônimo STRIDE.
Descrição	Detalhamento completo da ameaça.

Fonte: Autor.

No relatório final de um modelo de ameaças, a ferramenta lista as ameaças e descreve o que cada um significa. A definição de sugestão de mitigação da ameaça não é explicitamente estabelecida como parte do modelo da base de conhecimento.

2.3.3 Ferramenta Threat Dragon

A OWASP Threat Dragon (OWASP, 2020) é uma ferramenta visual de criação de modelos de ameaças sob a gestão do consórcio OWASP. Ela permite a criação visual de diagramas de fluxo de dados, a inserção das ameaças nos fluxos e a geração de relatórios automatizados.

Na ferramenta, cada elemento do diagrama de fluxo de dados é definido como um componente do sistema; as arestas representam o fluxo de transferência de dados entre os componentes. Cada componente e aresta é definido pelo par do tipo do componente e dos parâmetros associados. Os componentes representam os elementos de um DFD e são *actor, store, dataflow e process*. Os parâmetros possíveis são determinados a partir do tipo de componente, por exemplo, o componente *store* possui os parâmetros *store credentials, stores inventory, is encrypted e is signed.*

Seu modelo de automação é baseado em etiquetas, desta forma, utilizam-se palavras para descrever as características do comportamento de cada componente do modelo. Para a elicitação automática, utiliza-se uma consulta à base de conhecimento considerando a validação booleana do tipo de componente e dos parâmetros associados. Quando o tipo e os parâmetros do elemento do diagrama são iguais aos esperados na base de conhecimento, a ameaça é selecionada para que o usuário da ferramenta decida sobre a sua pertinência.

A base de conhecimento baseia-se no conjunto genérico de ameaças dos acrônimos STRIDE, LINDDUN e CIA. Adicionalmente, a ferramenta também considera as 21 ameaças listadas no OWASP Automated Threats to Web Applications (OWASP, 2019a), que é uma base de ameaças relacionadas ao acesso automatizado de aplicações. A Tabela 3 apresenta os campos da base de conhecimento.

Tabela 3 – Base de Conhecimento de Ameaças da *Threat Dragon*

Campo	**Descrição**
Condições	Conjunto de regras booleanas que expressam a pertinência ou não da ameaça para o componente avaliado.
Título	Nome que define a ameaça.
Tipo	Ameaça associada a um modelo de ameaças. Exemplo: S de *Spoofing* do STRIDE.
Tipo de Modelo	Tipo da classificação de ameaças utilizada. Exemplo: STRIDE, CIA, LINDDUN.
Status	Status do tratamento da ameaça.
Severidade	Classificação qualitativa da severidade.
Descrição	Detalhamento da ameaça.
Mitigação	Detalhamento dos controles de segurança para mitigar a ameaça.

Fonte: Autor.

A Tabela 3 apresenta a estrutura da Base de Conhecimento de Ameaças da ferramenta Threat Dragon, que organiza os principais atributos utilizados na modelagem e no tratamento de ameaças. Cada registro inclui um título que identifica a ameaça, seu tipo e o modelo de classificação adotado, como STRIDE, CIA ou LINDDUN. A tabela também contempla o status do tratamento, a severidade qualitativa do impacto e a descrição detalhada da ameaça. Além disso, o campo de condições define, por meio de regras booleanas, a aplicabilidade da ameaça ao componente avaliado, enquanto o campo mitigação apresenta as medidas de segurança recomendadas para reduzir ou eliminar o risco associado.

2.3.4 Ferramenta SPARTA

A ferramenta SPARTA foi desenvolvida pelo grupo de pesquisa DistriNet da KU Leuven, com o propósito de apoiar a análise integrada de riscos de segurança e privacidade (SION et al., 2018). Seu objetivo consiste em automatizar a geração de ameaças, aplicando critérios de priorização baseados em risco. Foi disponibilizado inicialmente em 2018 e foi evoluindo até a sua última versão em 2022 (LANDUYT et al., 2024).

A ferramenta possibilita a modelagem automática por meio de DFDs, de catálogo de ameaças e de padrões de soluções de segurança. As ameaças possíveis baseiam-se no conjunto STRIDE e LINDDUN, e as soluções de segurança são previamente definidas em um catálogo específico, contendo informações sobre a aplicabilidade do padrão e os controles de segurança associados.

A elicitação automática de ameaças é realizada com base na verificação de regras de relacionamento ou de elementos do diagrama de fluxo de dados. Cada tipo de ameaça que precisa ser elicitado contém uma lista de padrões de elementos do DFD a serem verificados. A sintaxe utilizada é a seguinte: [ElementType][To/Through][ElementType], com um "*" (asterisco) após uma das três partes para indicar a localização da ameaça. Como exemplo do uso desta sintaxe, se considerarmos uma ameaça relativa a um fluxo de dados que atravessa um limite de confiança (trust boundary), indo de uma Entidade Externa para um Processo, será expressa, conforme a Tabela 4 abaixo (KU Leuven, DistriNet Research Group, 2022).

Tabela 4 – Exemplo de expressões para elicitação de ameaças do SPARTA

ID	**Regra**	**Descrição**
1	ExternalEntity*ThroughProcess	Ameaça expressa no lado do envio da informação.
2	ExternalEntityThrough*Process	Ameaça expressa no próprio fluxo de dados.
3	ExternalEntityThroughProcess*	Ameaça expressa no lado do recebimento da informação.

Fonte: Adaptado de KU Leuven, DistriNet Research Group (2022).

Após a elicitação inicial das ameaças, o SPARTA realiza uma verificação se controles de segurança já foram previamente adotados pelo modelo, desta forma, filtrando ameaças para que aquelas que já foram tratadas não sejam novamente identificadas.

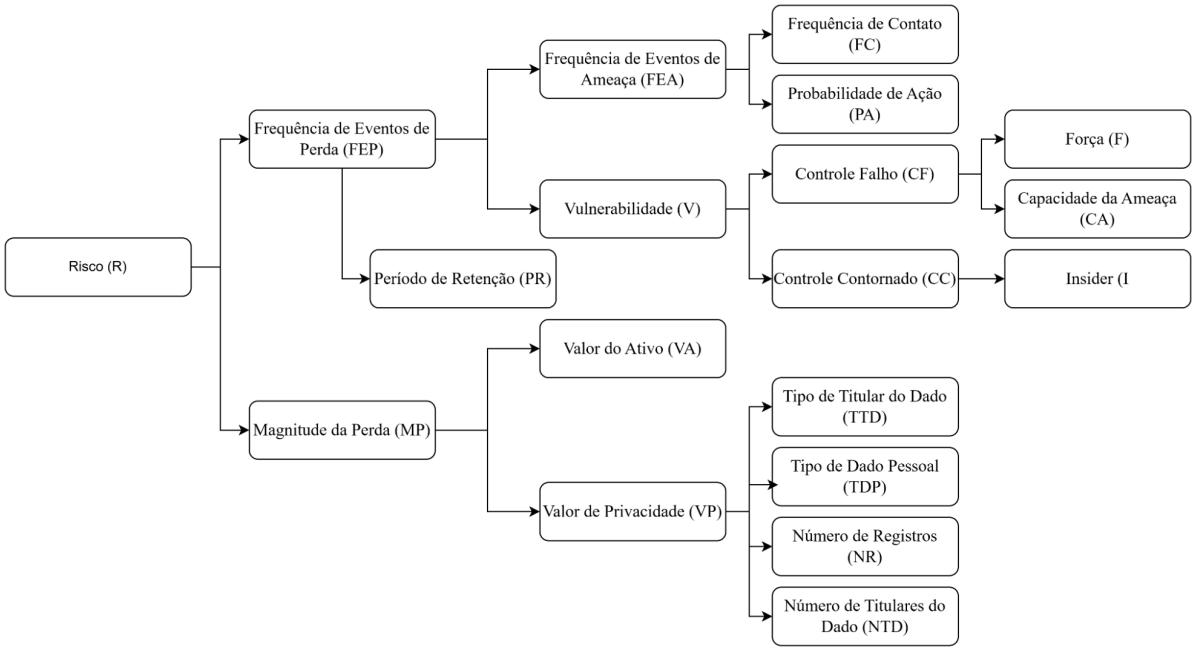
Para cada ameaça é calculado o risco associado utilizando uma metodologia quantitativa própria baseada no modelo de risco FAIR (SION et al., 2018). Para o cálculo são utilizados diversos componentes definidos nos elementos do DFD e nos catálogos de soluções, ameaças e atacantes definidos pelo *expert* de segurança da informação e pelos *stakeholders*. A Figura 9 abaixo apresenta os componentes utilizados no cálculo do risco.

O Risco (R) é calculado pela multiplicação entre a Magnitude da Perda (MP), correspondente ao valor anual do dano financeiro atribuído ao elemento do DFD ameaçado, e a Frequência de Eventos de Perda (FEP), que representa a taxa de ataques bem-sucedidos ou incidentes de privacidade.

A Frequência de Eventos de Perda (FEP) é determinada pela Frequência de Eventos de Ameaça (FEA), pelo Período de Retenção (PR) e pela Vulnerabilidade (V). A FEA expressa a frequência com que ocorrem tentativas de ataque, independentemente de serem bem-sucedidas, e é calculada a partir da Frequência de Contato (FC), que indica o quão frequentemente o atacante interage com o sistema, e da Probabilidade de Ação (PA), que representa a chance do atacante efetivamente realizar uma tentativa em cada contato.

A Vulnerabilidade (V) representa a probabilidade de uma ameaça de segurança ou privacidade se manifestar com sucesso. Seus atributos são definidos de modo a refletir a incerteza das estimativas realizadas por especialistas, utilizando os parâmetros mínimo, provável, máximo e nível de confiança da avaliação. A vulnerabilidade é obtida pelo maior valor entre o Controle Falho (CF) e o Controle Contornado (CC). O CF indica o grau em que um controle de segurança

Figura 9 – Componentes do Cálculo do Risco do SPARTA



Fonte: Autor adaptado de (KU Leuven, DistriNet Research Group, 2022)

ou privacidade pode ser superado por um atacante, sendo calculado a partir de uma função que relaciona a Força (F), isto é, a capacidade do controle de resistir a ataques, à Capacidade de Ameaça (CA), que expressa o nível de habilidade ou poder do atacante em vencê-lo. Já o CC representa a probabilidade de um atacante contornar um controle, independentemente de sua força, e esse fator está associado à atuação de insiders (I).

O Período de Retenção (PR) representa o tempo em que os dados pessoais são armazenados ou processados. Permitindo que as ameaças à privacidade sejam consideradas ao longo de sua existência.

A Magnitude da Perda (MP) é calculada com base no Valor do Ativo (VA) e no Valor da Privacidade (VP). O VA representa o prejuízo financeiro que a organização perde em caso de ataque bem-sucedido, enquanto o VP expressa o impacto sobre os titulares dos dados afetados. Esse último é determinado por uma função que considera o Tipo de Titular do Dado (TTD), o Tipo de Dado Pessoal (TDP), o Número de Registros (NR) e o Número de Titulares de Dados (NTD).

Ao final do processo de enriquecimento das informações do diagrama de fluxo de dados, é possível realizar a elicitação automática das ameaças e do cálculo dos riscos. A base de conhecimento da ferramenta é dividida no catálogo de ameaças e no catálogo de soluções de segurança. A Tabela 5 descreve os campos disponíveis.

Considerando a Tabela 5 , o catálogo de ameaças representa todas as ameaças à segurança e à privacidade que podem ser detectadas pela ferramenta. O catálogo de padrões define as

Tabela 5 – Base de Conhecimento de Ameaças da SPARTA

Artefato	**Campo**	**Descrição**
Catálogo de Ameaças	Nome	Identificador da ameaça.
Catálogo de Ameaças	Título	Tipo do elemento do diagrama.
Catálogo de Ameaças	Descrição	Descrição em alto nível da ameaça.
Catálogo de Ameaças	Comentários	Detalhamento em linguagem natural da ameaça.
Catálogo de Ameaças	Habilitado	Definição se deve ser considerado ou não pelo modelo.
Catálogo de Ameaças	Condição	Lista de regras para identificação da ameaça no DFD.
Catálogo de Padrões	Nome	Nome do padrão (ex.: *Secure Pipe*).
Catálogo de Padrões	Descrição	Detalhamento do padrão de segurança.
Catálogo de Padrões	Papel	Definição de onde o padrão pode ser aplicado no DFD.
Catálogo de Padrões	Contramedida	Controles associados ao padrão com definição da dificuldade da ameaça mitigada.

Fonte: Autor.

soluções de segurança possíveis para uso e suas escolhas no diagrama de fluxo de dados afetam o resultado da elicitação das ameaças (SION et al., 2018).

2.3.5 Ferramenta PyTM

A ferramenta Pythonic Threat Modeling (OWASP, 2019c) é uma ferramenta que realiza a modelagem de ameaças como código. Assim como a OWASP Threat Dragon, a PyTM também é uma ferramenta sob a gestão do consórcio OWASP.

Diferente das demais abordagens, esta ferramenta disponibiliza um conjunto de classes (Datastore, Process, External Entity, Dataflow, Boundary) na linguagem de programação Python para descrever os relacionamentos entre os elementos que compõem o diagrama de fluxo de dados que representa o modelo de ameaças. Adicionalmente, algumas classes são disponibilizadas como Server, Actor e Lambda, elas são especializações das classes básicas de um diagrama de fluxo de dados e representam, respectivamente, servidor, ator do sistema e funções lambda. Cada uma das classes é constituída de atributos que definem características de cada elemento, como, por exemplo, na classe Datastore, são definidos os atributos *isSql*, que indica se o armazenamento de dados é um banco de dados SQL, e o atributo *isHardened*, que indica se o armazenamento de dados já passou por um processo de melhoria de segurança. Estes atributos serão utilizados para auxiliar o mecanismo de automação na elicitação das ameaças.

A Tabela 6 apresenta um exemplo do uso da linguagem de programação Python na descrição de um modelo de ameaças.

Tabela 6 – Exemplo de Código do uso do PyTM

Código

```
User_Web = Boundary("User/Web")
Web_DB = Boundary("Web/DB")
user = Actor("User")
user.inBoundary = User_Web
web = Server("Web Server")
web.OS = "CloudOS"
web.isHardened = True
web.sourceCode = "server/web.cc"
db = Datastore("SQL Database (*)")
db.OS = "CentOS"
db.isHardened = False
db.inBoundary = Web_DB
db.isSql = True
db.inScope = False
db.sourceCode = "model/schema.sql"
comments = Data(
  name="Comments",
  description="Comments in HTML or Markdown",
  classification=Classification.PUBLIC,
  isPII=False,
  isCredentials=False,
  credentialsLifeTime.LONG,
  isStored=True,
  isSourceEncryptedAtRest=False,
  isDestEncryptedAtRest=True
)
results = Data(
  name="results",
  description="Results of insert op",
  classification=Classification.SENSITIVE,
  isPII=False,
  isCredentials=False,
  credentialsLifeTime.LONG,
  isStored=True,
  isSourceEncryptedAtRest=False,
  isDestEncryptedAtRest=True
)
my_lambda = Lambda("CleanDBevery6hours")
```

Continuação na próxima página

Código (continuação)

```
my_lambda.hasAccessControl = True
my_lambda.inBoundary = Web_DB
my_lambda_to_db = Dataflow(my_lambda, db, "(lambda):Periodically
cleans DB")
my_lambda_to_db.protocol = "SQL"
my_lambda_to_db.dstPort = 3306
user_to_web = Dataflow(user, web, "User enters comments (*)")
user_to_web.protocol = "HTTP"
user_to_web.dstPort = 80
user_to_web.data = comments
web_to_user = Dataflow(web, user, "Comments saved (*)")
web_to_user.protocol = "HTTP"
web_to_db = Dataflow(web, db, "Insert query with comments")
web_to_db.protocol = "MySQL"
web_to_db.dstPort = 3306
web_to_db.data = comments
db_to_web = Dataflow(db, web, "Comments contents")
db_to_web.protocol = "MySQL"
db_to_web.data = results
tm.process()
```

Fonte: (OWASP, 2019c).

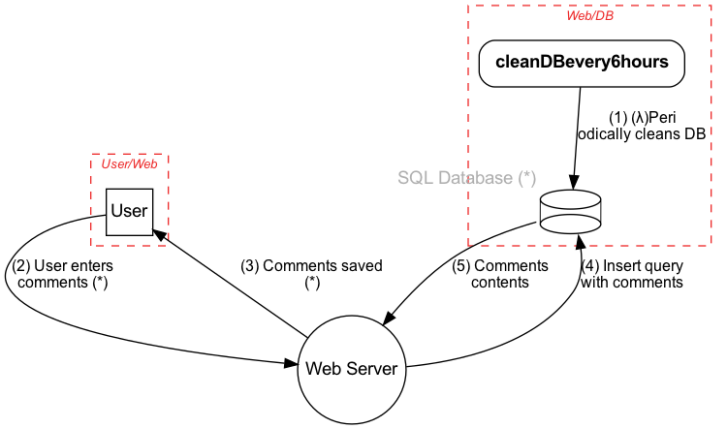
A descrição do diagrama de fluxo de dados apresenta um processo servidor web (web server) que disponibiliza e recebe comentários a partir de uma entidade externa usuário (*user*), e armazena e consulta o seu conteúdo em um banco de dados (SQL Database). Adicionalmente, um processo lambda (*cleanDBevery6hours*) executa uma limpeza no banco de dados. A Figura 10 apresenta o diagrama gerado a partir do código-fonte descrito na Tabela 6.

Para a elicitação automática de ameaças, a ferramenta realiza a navegação nos relacionamentos do modelo de ameaças considerando, na base de conhecimento, o alvo da ameaça e a condição de aplicabilidade da ameaça. O alvo da ameaça é definido como o tipo de classe do elemento do diagrama do fluxo de dados, como, por exemplo, *Process ou Lambda*. A condição de aplicabilidade é uma expressão booleana que utiliza as propriedades das classes como atributos para verificar a veracidade.

Por exemplo, considere a ameaça *Session Sidejacking*, que consiste no roubo de sessão de um usuário autenticado por meio da interceptação de tokens de sessão em trânsito. Neste caso, os alvos potenciais são os elementos *Server* e *Dataflow*, pois a exploração depende tanto da vulnerabilidade na comunicação entre cliente e servidor quanto da maneira como o servidor gerencia as sessões. A condição lógica (expressão booleana) que define a suscetibilidade a essa ameaça é composta pelos seguintes critérios:

- O atributo protocol deve ser igual a HTTP, ou o atributo *useVPN* deve ser falso, indicando que a

Figura 10 – Diagrama de Fluxo de Dados gerado pela PyTM



Fonte: (OWASP, 2019c)

comunicação ocorre sem criptografia ou fora de um túnel seguro;

- Adicionalmente, o atributo useSessionTokens deve ser verdadeiro, o que indica que o sistema utiliza tokens de sessão para autenticação e, portanto, está sujeito à interceptação desses identificadores.

Esse conjunto de condições representa a avaliação que a ferramenta realiza para indicar que existe essa ameaça no modelo. Durante a navegação do modelo, caso estes parâmetros sejam considerados válidos, a ameaça será listada. A Tabela 7 descreve todos os campos disponíveis na base de conhecimentos da ferramenta.

Tabela 7 – Base de Conhecimento de Ameaças da PyTM

Campo	**Descrição**
SID	Identificador da ameaça.
Alvo	Tipo do elemento do diagrama.
Descrição	Descrição em alto nível da ameaça.
Detalhes	Detalhamento em linguagem natural da ameaça.
Probabilidade de Ataque	Classificação da probabilidade de ocorrência.
Severidade	Severidade do impacto.
Condição	Regra booleana que descreve a aplicabilidade da ameaça.
Pré-requisitos	Exigências em linguagem natural para ocorrência da ameaça.
Mitigações	Controles que podem ser utilizados para deter a ameaça.
Exemplos	Exemplos de ocorrência da ameaça.
Referências	Referências a CAPEC e CWE.

Fonte: Autor.

A Tabela 7 apresenta as informações utilizadas pela ferramenta para identificação e avaliação de ameaças. Cada registro contém um identificador da ameaça (SID), alvo correspondente no diagrama e uma descrição em alto nível, complementada por detalhes em linguagem natural. Incluem-se também campos para probabilidade de ataque, severidade do impacto e uma condição booleana, que define quando a ameaça é aplicável. Além disso, são descritos os pré-requisitos necessários para sua ocorrência, as mitigações recomendadas, exemplos práticos e referências a bases externas como CAPEC e CWE, que aprofundam o entendimento técnico de cada ameaça.

2.3.6 Ferramenta Threat-agile

A ferramenta Threatagile foi desenvolvida por Threatagile (2020) como uma ferramenta de modelagem de ameaças baseada na definição de arquivos YAML (*YAML Ain't Markup Language*) (YAML, 2001) do diagrama de fluxo de dados que representa a arquitetura, seus ativos e relacionamentos. Seu funcionamento é feito através de linha de comando e sua execução pode ser realizada com ferramentas de integração contínua. Embora tenha essa possibilidade, sua análise é individualizada, não permitindo acompanhar a evolução do modelo de ameaças para as múltiplas versões de um sistema.

Ela analisa o modelo de ameaças definido no YAML como um grafo de componentes conectados e gera alguns elementos: os gráficos do modelo, os potenciais riscos e ameaças, as sugestões de controles de segurança e os relatórios para os times de desenvolvimento. Os elementos do diagrama de fluxo de dados são definidos como ativos de dados e ativos técnicos. Os ativos de dados são enriquecidos com os atributos de descrição, uso, tags, origem, dono do ativo, quantidade (definição qualitativa), confidencialidade, integridade e disponibilidade. Nos ativos técnicos são definidos: a) o nome (ex. Apache Webserver); b) a descrição; c) o tipo de elemento no diagrama de fluxo de dados; d) o uso, se é utilizado por humanos; e) se é parte do escopo de avaliação; f) o tamanho; g) a tecnologia; h) tags; i) se está na internet; j) o tipo de máquina (física, virtual, cloud, etc); k) suporte a criptografia; l) o dono; m) a confidencialidade, n) a integridade; o) a disponibilidade; p) se é multitenant; q) se é redundante e, r) se possuí partes desenvolvidas por customização.

A elicitação automática de ameaças baseia-se nas informações sobre os elementos do diagrama de dados e os atributos preenchidos. Para cada fluxo do DFD, as regras de seleção da ameaça são aplicadas e, caso o retorno da aplicabilidade seja positivo, elas são incluídas na lista de ameaças. As regras são fixas e são ao todo 42 regras para seleção de ameaças. Elas são implementadas na linguagem de programação GO (The Go Authors, 2025).

Após a seleção das ameaças, os dados da base de conhecimento de ameaças são selecionados e apresentados na forma de relatório para o usuário final. A Tabela 8 abaixo apresenta a lista de atributos disponíveis na base de conhecimento da ferramenta.

A Tabela 8 apresenta os principais campos da base de conhecimento e suas respectivas funções. Cada registro contém informações essenciais sobre uma ameaça, incluindo seu identificador (ID), o alvo no diagrama, a descrição e o impacto associado. São também incluídas referências a controles reconhecidos, como o OWASP ASVS, Cheat Sheets e CWE, bem como instruções práticas de ação, mitigação e verificação. Além disso, a base contempla campos voltados à classificação do risco, lógica de detecção, avaliação de risco e identificação de falsos positivos, oferecendo ainda exemplos de reações em possíveis falhas.

2.3.7 Ferramenta Threat Manager Studio

O Threats Manager Studio foi desenvolvido por TMS (2020) como uma solução extensível e orientada à automação, com suporte a extensões para geração automática de eventos de ameaça e associação de mitigações.

A elicitação automática de uma ameaça é baseada na decisão lógica de aplicação ou não com

Tabela 8 – Base de Conhecimento de Ameaças da Threat-agile

Campo	**Descrição**
ID	Identificador da ameaça.
Alvo	Tipo do elemento do diagrama.
Descrição	Descrição em alto nível.
Impacto	Descrição textual do impacto.
ASVS	Referências ao OWASP ASVS.
CheatSheet	Link para controles OWASP.
Ação	Ação recomendada para mitigação.
Mitigação	Descrição do controle de mitigação.
Verificação	Questões de verificação da mitigação.
Função	Categoria do risco.
STRIDE	Mapeamento STRIDE.
Lógica de Detecção	Como identificar a ameaça.
Avaliação de Risco	Avaliação textual do risco.
Falsos Positivos	Orientações para identificação.
Reação em possível falha	Exemplos de ocorrência.
CWE	Referência ao CWE.

Fonte: Autor.

base na avaliação de um conjunto de regras estruturadas sequencialmente, como um algoritmo, onde são avaliadas:

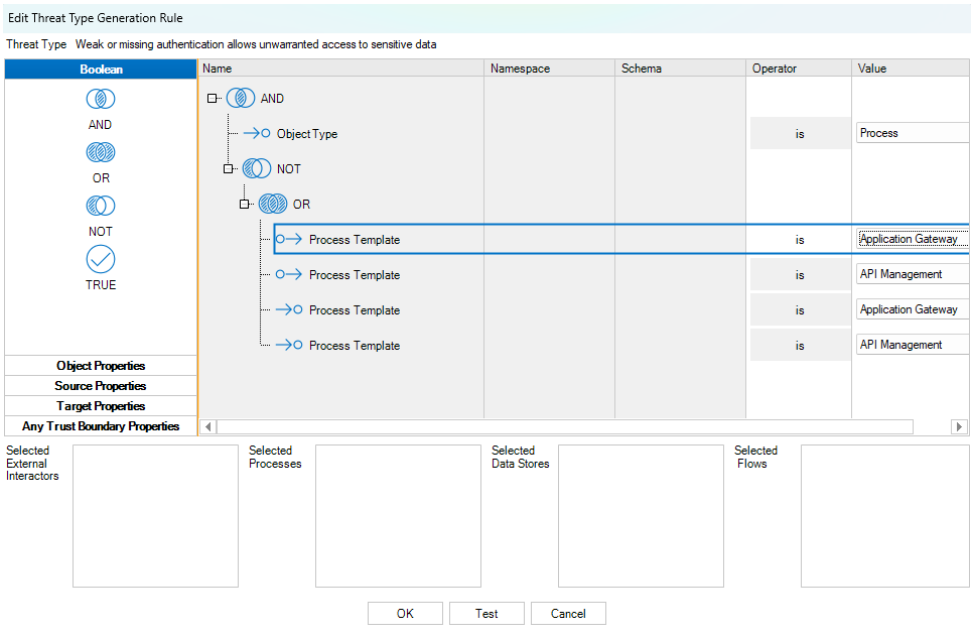
- condições booleanas;
- condições associadas às propriedades dos atores, dos processos, do armazenamento de dados, do fluxo de dados e dos limites de confiança;
- condições associadas às propriedades do elemento de origem de fluxo de dados;
- condições associadas às propriedades do elemento de destino de fluxo de dados;
- de propriedades dos limites de confiança.

Estas regras são definidas de forma visual e na forma de árvore. Quando as condições são avaliadas como positivas, a ameaça é selecionada e compõe a lista de ameaças. A Figura 11 apresenta a interface gráfica para a definição das regras de seleção de ameaças. A interface gráfica apresentada na Figura 11 apresenta ao lado esquerdo as condições possíveis de serem criadas. Ao meio está definida a sequência que será avaliada para indicar a presença ou não de uma ameaça.

Após a definição das regras para elicitação das ameaças ou o uso de um catálogo previamente definido pela ferramenta, o diagrama de fluxo de dados é avaliado e tem sua lista completa de ameaças definida. Para cada ameaça, são definidos os dados presentes na base de conhecimento para descrever, com maior profundidade, as ameaças identificadas. A Tabela 9 apresenta a lista de atributos da base de conhecimento da ferramenta.

A Tabela 9 apresenta os campos que organizam informações essenciais para a análise e mitigação de riscos. Cada registro inclui o nome e a descrição da ameaça, sua severidade e as mitigações padronizadas disponíveis no catálogo. Também são descritos o tipo de controle de mitigação, classificado

Figura 11 – Interface em *low code* para definição de regras de seleção de ameaças



Fonte: O Autor

Tabela 9 – Base de Conhecimento de Ameaças da Threat Manager Studio

Campo	**Descrição**
Nome	Nome da ameaça.
Descrição	Detalhamento textual da ameaça.
Severidade	Escala de não definido até crítico.
Mitigações Padronizadas	Soluções do catálogo para mitigação.
Descrição da Mitigação	Detalhamento textual da mitigação.
Tipo de Controle	Preventivo, Detectivo ou Recuperação.
Força do Controle	Negligenciado, Fraco, Médio, Forte ou Máximo.

Fonte: Autor.

como preventivo, detectivo ou de recuperação, e a força do controle, avaliada em níveis que variam de negligenciado a máximo.

2.3.8 Ferramenta Sla-generator

A ferramenta utiliza um modelo específico baseado em grafos para descrever as interações de um sistema (GRANATA; RAK; SALZILLO, 2022). Cada nó representa um componente e as arestas representam o relacionamento entre eles. Cada nó possui duas etiquetas pré-definidas que classificam o tipo de ativo de software que o nó representa. A primeira etiqueta classifica o tipo de ativo de software de forma mais genérica, e se necessário a segunda etiqueta faz esta classificação de forma mais específica. Por exemplo, em um modelo que precise descrever um nó que representa uma infraestrutura de nuvem pública de servidores, a primeira etiqueta seria service e a segunda etiqueta seria IaaS. Cada nó também possui um grupo de propriedades que são utilizadas para especificar melhor sua definição. A propriedade mandatória é o tipo de ativo (*asset type*) que define o comportamento do ativo representado pelo nó. Os tipos de ativos possíveis são determinados pelas etiquetas selecionadas para o nó. As arestas que definem

os relacionamentos entre os nós são marcadas pelas etiquetas de *uses, hosts, provides e connects*.

A base de conhecimento das ameaças é representada por oito campos, conforme detalhado na Tabela 10. Ela também apresenta os campos que organizam as informações utilizadas para identificar e classificar ameaças nos modelos de comunicação. Cada registro inclui o título da ameaça, o tipo de ativo envolvido, o relacionamento entre os componentes e o protocolo de comunicação utilizado. Também são descritos o papel no relacionamento, o comportamento detalhado da ameaça, sua classificação STRIDE e o elemento comprometido, que indica se o impacto ocorre no próprio ativo, na origem ou no destino da comunicação.

Tabela 10 – Base de Conhecimento de Ameaças da *Sla-generator*

Campo	**Descrição**
Ameaça	Título da ameaça.
Tipo de Ativo	Tipo do ativo representado pelo nó.
Relacionamento	Etiqueta do tipo de relacionamento.
Protocolo	Protocolo usado na comunicação.
Papel no relacionamento	Papel na comunicação.
Comportamento	Descrição detalhada da ameaça.
STRIDE	Classificação STRIDE.
Comprometido	Ativos ou comportamento comprometido.

Fonte: Autor.

Seu método de automação baseia-se em etiquetas, relacionamentos e propagação. Além de buscar palavras que definam características dos componentes do modelo, também são verificadas as interações entre os componentes e se uma determinada ameaça se propaga para outros componentes vizinhos do modelo, por meio da indicação do comprometimento existente na base de conhecimento. Por exemplo, na elicitação automática são utilizados o tipo de ativo e o protocolo de comunicação, desta forma, para cada nó do grafo são consultadas na base de conhecimento as ameaças que se enquadram de acordo com as etiquetas selecionadas. Caso a ameaça comprometa apenas o ativo avaliado, ele será relacionado à ameaça identificada.

2.4 Comparativo com as ferramentas avaliadas

No contexto das ferramentas de modelagem de ameaças automatizadas, (SHI et al., 2021) sugeriu um conjunto de oito características para avaliar ferramentas de modelagem de ameaças, a saber: tipo de modelo textual ou gráfico, reusabilidade do modelo, bibliotecas de ameaças, avaliação de ameaças, sugestão de mitigação, disponibilidade, maturidade de desenvolvimento e integração com o ciclo de vida do desenvolvimento de software. Já Granata e Rak (2023) aborda quatro características: tipo de modelo textual ou gráfico, domínio de utilização, suporte à análise de risco e método de elicitação de ameaças.

Como o foco deste trabalho está relacionado com o uso de técnicas para a elicitação automática de ameaças, dos controles de segurança e das bases de conhecimento, a comparação da ferramenta Threat Copilot com os trabalhos relacionados foi feita utilizando as seguintes características: tipo de modelo, biblioteca de ameaças, sugestão de mitigação e método de elicitação automática. A Tabela 11 abaixo apresenta a avaliação das ferramentas de modelagem de ameaças automatizadas.

Tabela 11 – Ferramentas de Código Aberto para Modelagem de Ameaças Automatizada

Ferramenta	**Tipo de Modelo**	**Base de Conhecimento**	**Sugestão de Mitigação**	**Método de Elicitação**
AutSec	Gráfico	Própria	Sim, baseado em risco	Etiquetas / Relacionamento
Microsoft TMT	Gráfico	STRIDE	Não	Relacionamento
OWASP Threat Dragon	Gráfico	STRIDE / LINDDUN / CIA	Sim	Etiquetas
SPARTA	Gráfico	STRIDE	Sim, baseado em risco	Relacionamento
PyTM	Textual / Código	CAPEC / CWE	Sim	Etiquetas
Threat-agile	Textual / Gráfico	Própria / STRIDE	Sim	Etiquetas / Relacionamento
Threat Manager Studio	Gráfico	Própria / STRIDE	Sim	Etiquetas / Relacionamento
Sla-generator	Gráfico	Própria	Sim	Etiquetas / Relacionamento / Propagação

Fonte: Autor.

Os trabalhos listados representam as diversas abordagens e técnicas para a elicitação automática de ameaças e das bases de conhecimento envolvidas. No contexto de sistemas de recomendação para cibersegurança, Pawlicka et al. (2021) realizaram uma revisão sistemática da literatura e Ferreira, Silva e Uriarte (2023) também realizaram um survey, em ambas as pesquisas, também não se identificou uma abordagem do uso de sistemas de recomendação para modelagem de ameaças conforme os tipos possíveis definidos por Aggarwal (2016). Desta forma, este trabalho se diferencia dos demais principalmente por tratar o problema da elicitação automática de ameaças como um sistema de recomendação baseado em conteúdo. Neste sentido, comparando com os demais trabalhos de elicitação de ameaças automatizadas, tem-se as seguintes características:

- Tipo de modelo: abordagem híbrida, sendo textual na linguagem de descrição YAML (YAML, 2001) e gráfico por suportar a conversão do modelo gráfico para o modelo textual;
- Base de conhecimento: base própria, estruturada a partir de modelos de ameaças previamente produzidos no contexto da Organização X;
- Sugestão de mitigação: previsão de recomendação de controles de segurança associados às ameaças identificadas;
- Método de elicitação automática: abordagem fundamentada em etiquetas, relacionamento e similaridade. Os componentes do diagrama de fluxo de dados do modelo de ameaças são rotulados por etiquetas, que são comparadas por meio de uma hierarquia semântica de termos. Em termos gerais, a proposta considera os rótulos dos elementos do diagrama de fluxo de dados, suas relações estruturais e o grau de proximidade entre fluxos de dados analisados e registros presentes na base de conhecimento.

Adicionalmente, uma nova característica também foi considerada:

- Dinamicidade da base de conhecimento: a proposta prevê uma base de conhecimento evolutiva, capaz de ser ampliada com a incorporação de novos modelos de ameaças, em contraste com abordagens fundamentadas em bases estáticas.

Em conjunto, essas características diferenciam a proposta em relação aos trabalhos correlatos e evidenciam seu posicionamento como uma abordagem orientada ao reaproveitamento de conhecimento histórico de modelagens anteriores. Os dessa proposta são apresentados no capítulo seguinte.

3 O SISTEMA DE RECOMENDAÇÃO *THREAT COPILOT*

Este capítulo apresenta a concepção e o funcionamento da ferramenta *Threat Copilot*, um sistema de recomendação desenvolvido nesta pesquisa para apoiar a elicitação automática de ameaças durante a modelagem de ameaças. São detalhados os três pilares do sistema de recomendação proposto: a elaboração do dataset que constitui a base de conhecimento da ferramenta, a arquitetura do sistema de recomendação e a heurística de recomendação das ameaças mais similares. Por fim, são apresentados detalhes sobre a implementação da ferramenta.

3.1 Elaboração do dataset de modelos de ameaças

O resultado dos processos de modelagem de ameaças comumente são diagramas e documentos que listam quais as ameaças identificadas e os controles de segurança que devem ser aplicados no contexto do produto sob avaliação. Como existem diversos métodos e ferramentas para a modelagem, também é comum que esses modelos sejam armazenados em formatos muitas vezes não estruturados, como, por exemplo, fotos de diagramas de fluxo de dados elaboradas fisicamente em quadros brancos ou em arquivos textuais gerados por ferramentas de edição de texto. No caso de ferramentas, cada solução possui seu formato final de armazenamento, seja os arquivos no formato JSON gerados pela a OWASP Threat Dragon, ou os arquivos XML gerados pela Microsoft TMT ou até mesmo código-fonte na linguagem de programação Python gerado pela ferramenta PyTM.

Associada a pluralidade de formatos, tem-se a baixa disponibilidade de modelos de ameaças reais públicos (PAIDY, 2023; WU et al., 2025). Naturalmente, o resultado dos modelos de ameaça contém dados sensíveis que podem ser utilizados para facilitar a construção de um ataque bem-sucedido por agentes mal-intencionados. Algumas iniciativas de catalogação de modelos reais de ameaças foram identificadas, como:

- OWASP Threat Cookbook (OWASP, 2019b) e Threat Examples (ELIYAHU, 2022) Apresentam modelos que não seguem um padrão comum e não estão descritos em formato estruturado, o que dificulta sua utilização em estudos comparativos ou reprodutíveis.
- Tuma et al. (2020) apresentou 26 diagramas de fluxo de dados correspondentes a quatro sistemas fictícios, elaborados a partir de um experimento conduzido com desenvolvedores e especialistas. Embora o estudo não esteja diretamente relacionado ao processo tradicional de modelagem de ameaças, os diagramas foram empregados para identificar e analisar falhas de design de segurança.
- Wu et al. (2025) definiu 50 modelos de ameaças no contexto do setor bancário na ferramenta Microsoft Threat Modeling Tool. As ameaças foram identificadas automaticamente pela ferramenta e os controles de segurança foram definidos manualmente a partir do NIST 800-53 (NIST, 2016). Este pesquisador não localizou nas referências do trabalho de Wu et al. (2025) e em consultas nos buscadores, o conteúdo dos modelos utilizados, o que limita seu uso em pesquisas empíricas e na construção de bases de conhecimento abertas.

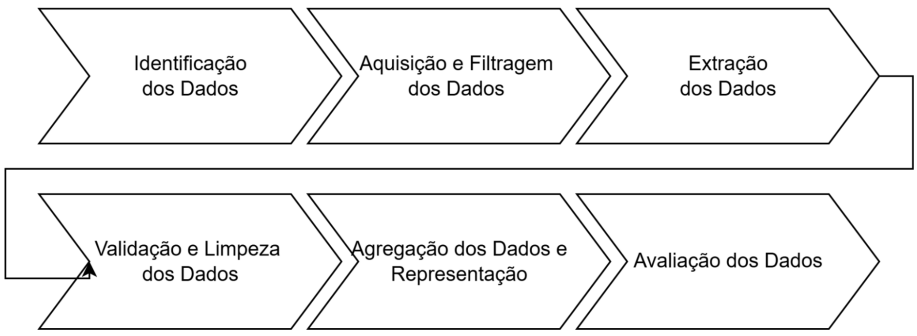
Diante da impossibilidade de utilizar modelos publicamente disponíveis, esta dissertação buscou mapear modelos de ameaças de forma estruturada e padronizada, convertendo representações não estruturadas em um formato comum, de modo a viabilizar sua utilização como base de conhecimento para o desenvolvimento e avaliação de um sistema de recomendação capaz de automatizar a elicitação de ameaças a partir do reúso de modelos previamente existentes na Organização X. O dataset construído ficou restrito a um recorte temporal e não poderá ser disponibilizado publicamente, pois é classificado como sigiloso e vinculado, de acordo com a norma de classificação da informação da Organização X.

3.1.1 Estágios de criação do dataset

Conforme (ERL; KHATTAK; BUHLER, 2016), o ciclo de vida para construção de datasets de Big Data é composto por nove estágios que organizam as atividades e tarefas relacionadas à aquisição, ao processamento, à análise e à transformação dos dados para consumo.

O volume de informação disponível na Organização X não se enquadra como algo de grande volume. Portanto, o dataset resultante desta dissertação não é considerado como *big data*. Além disso, o conjunto de informações não será utilizado para responder a hipóteses voltadas a uma necessidade de negócio, mas para ser utilizado como uma base de conhecimento para consulta e recomendação. Desta forma, os três estágios que envolvem a avaliação do caso de negócio, visualização dos dados e análise e interpretação dos resultados não foram utilizados. Assim esta pesquisa, adotou seis desses estágios: Identificação dos Dados; Aquisição e Filtragem; Extração; Validação e Limpeza; Agregação e Representação; Avaliação dos Dados. A Figura 12 apresenta os estágios utilizados no processo de construção do dataset de modelos de ameaças.

Figura 12 – Estágios utilizados na criação do dataset



Fonte: Adaptado de (ERL; KHATTAK; BUHLER, 2016)

As primeira subseção descreve a criação do dataset considerando os estágios de identificação dos dados, aquisição e filtragem dos dados, extração dos dados, validação e limpeza dos dados. A duas subseções posteriores apresentam as linguagens de descrição criadas para representar os modelos de ameaças e vocabulários. Por fim, a última subseção apresenta a agregação dos dados de ameaças e a avaliação dos dados obtidos. Esta divisão em subseções foi escolhida com o objetivo de tornar fluida a compreensão do trabalho realizado.

3.1.2 Criação do dataset

O escopo da coleta de dados abrangeu os modelos de ameaças elaborados pela área de desenvolvimento seguro da Organização X. Em 2019, como parte das ações previstas no plano estratégico da empresa, foram conduzidas iniciativas voltadas ao fortalecimento do ciclo de desenvolvimento seguro de software.

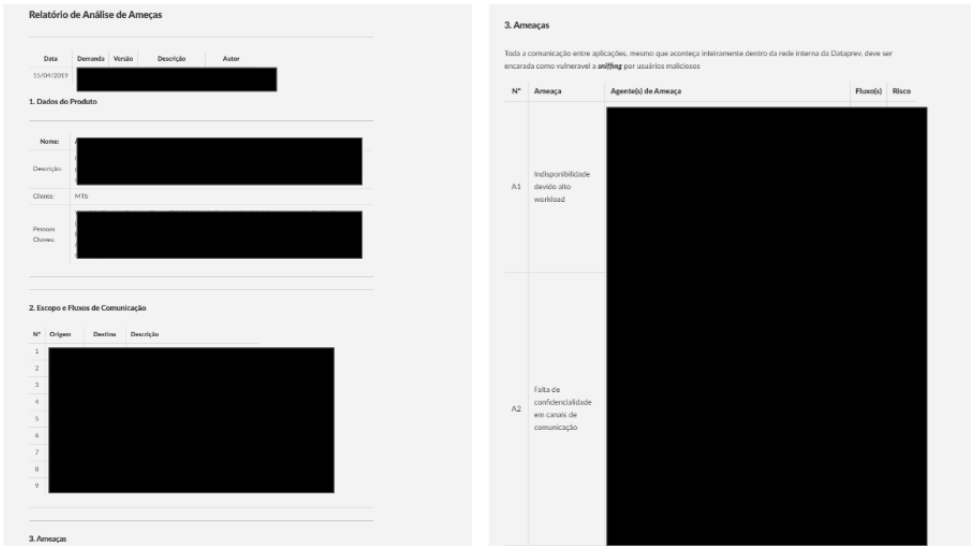
Durante este estágio, foram identificados e analisados os modelos de ameaças produzidos por sete analistas de segurança da informação, representando o conjunto de artefatos utilizados nesta pesquisa.

Este pesquisador identificou no repositório 69 produtos, dos quais 44 possuem modelos de ameaças descritos em formato textual na linguagem de texto Markdown (GRUBER; SWARTZ, 2004). A coleta dos dados seguiu alguns critérios de inclusão: está em um formato legível e o escopo do modelo de ameaças é um sistema de software do tipo aplicação web, batch ou mobile. Para a filtragem dos dados, os critérios de exclusão foram: não estar em meio digital, tipo de aplicação de business intelligence, modelo duplicado, modelo incompleto, modelo incoerente e modelo sem diagrama de fluxo de dados.

No total, 10 modelos foram eliminados, sendo 5 modelos excluídos por serem do tipo de aplicação de business intelligence e outros 5 modelos excluídos por estarem incompletos. Ao final do processo de coleta e filtragem dos dados, foram selecionados 34 modelos de ameaças restantes.

A Figura 13 apresenta um exemplo de relatório de modelagem de ameaças pertencente a esse conjunto de dados.

Figura 13 – Exemplo de Relatório de Modelagem de Ameaças avaliado



Fonte: O Autor

Considerando o exemplo da Figura 13, os relatórios coletados, normalmente, possuem a estrutura com dados da emissão do relatório, dados do produto, escopo e planos de comunicação e lista de ameaças. A lista de ameaças tem a descrição da ameaça e os fluxos no quais as ameaças estão presentes.

Por fim, ao término da execução do protocolo, na etapa de disponibilização dos dados, os modelos de ameaças descritos por meio da linguagem de descrição de modelos de ameaças adotada, descrita na

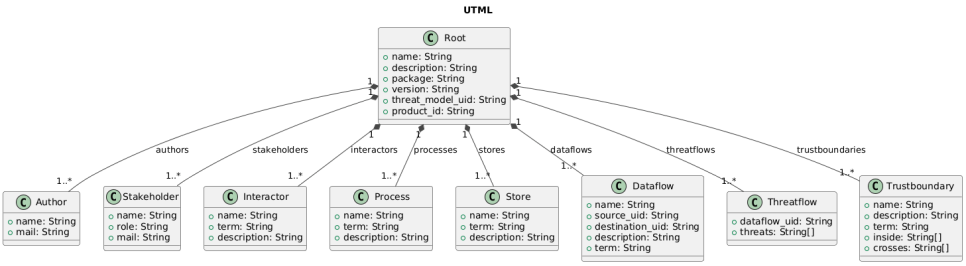
próxima subseção, foram armazenados em um repositório de versionamento da organização, garantindo seu uso e rastreabilidade em futuras atividades.

3.1.3 Linguagem de descrição do modelo de ameaças

Para a conversão dos modelos, foi especificada uma linguagem de descrição de ameaças denominada de UTML (*Unified Threat Modeling Language*) no formato YAML (YAML, 2001). Sua escolha deve-se pelo fato do YAML apresentar uma sintaxe mais intuitiva e legível para o humano quando comparado aos formatos JSON e XML (ERIKSSON; HALLBERG, 2011), também tendo sido a linguagem utilizada para para descrever cenários legíveis de simulação de ambientes em nuvem (FILHO; RODRIGUES, 2014) e simplificação da *Systems Biology Markup Language* (VANHOEFER et al., 2021).

A UTML visa descrever, de forma simplificada e textual, os modelos de ameaças que utilizam a notação gráfica de diagramas de fluxo de dados para representar os elementos e relacionamentos de um sistema. Ela também permite a categorização através da definição de termos para os componentes e fluxos de dados. Recentemente, também foi publicado um outro modelo de descrição de modelos de ameaças baseado em YAML denominado de *OpenThreatModel* (IriusRisk, 2023). Espera-se que a escolha do formato estruturado de representação do modelo de ameaças não interfira no funcionamento do sistema de recomendação, apenas facilite ou não a sua manipulação. A Figura 14 apresenta de forma visual um exemplo de uma definição de modelo de ameaças usando a UTML.

Figura 14 – Linguagem de descrição UTML



Fonte: O Autor

Na Figura 14 são definidos alguns tipos de dados para descrever um diagrama de fluxo de dados para um modelo de ameaças:

- **Root:** Define os atributos do modelo de ameaças. O atributo package é relevante para identificar de forma única os elementos do diagrama de fluxo de dados e sua forma de descrição segue o modelo de pacotes da linguagem JAVA. Exemplo: “br.gov.empresa.meuproduto”. Todos os elementos (processos, atores, fluxos de dados, armazenamento e limites de confiança) são referenciados unicamente com a junção do pacote mais o nome do elemento. Exemplo: “br.gov.empresa.meuproduto.Usuario”.
- **Author e Stakeholder:** Definem os responsáveis pela criação do modelo de ameaças.
- **Interactor, Process, Store:** Definem os elementos do diagrama de fluxo de dados. Cada elemento possui um nome, o termo que define o elemento no vocabulário e uma descrição.

- **Dataflow:** Define os relacionamentos entre os elementos. Além do nome, do termo e da descrição, são definidos a origem do fluxo de dados e o destino. Estes atributos são definidos a partir da identificação única do elemento. Por exemplo: "source_uid: br.gov.empresa.meuproduto.Usuario".
- **Threatflow:** Define a lista de ameaças selecionadas para um fluxo de dados. O atributo element_uid identifica o elemento do fluxo de dados ao qual a ameaça está relacionada. As ameaças são referenciadas através dos seus identificadores da base de conhecimento de ameaças, como, por exemplo, "DTP-1".
- **Trustboundary:** Define os limites de confiança do diagrama de fluxo de dados. Além do nome, do termo e da descrição, são definidos quais fluxos de dados cruzam o limite de confiança e quais elementos estão internos ao limite de confiança.

Para validar a capacidade de descrição da UTML, foi elaborado um exemplo que utiliza todas as características da referida linguagem. Este exemplo utilizou como base um dos modelos de ameaças convertidos. A descrição do exemplo está anonimizada e pode ser vista na Tabela 12.

Tabela 12 – Exemplo de Modelo de Ameaças na UTML

Código UTML

```
--
name: PRODUTOB
description: Anonimizado
package: br.produtos
authors:
  - name: Marcos Random
    mail: marcos.random@gmail.com
stakeholders:
  - name: Pedro Aleatorio
    role: Gerente de Projeto
    mail: pedro.aleatorio@gmail.com
version: 1.0
threat_model_uid: br.produtos.produtob.tm
product_id: br.produtos.produtob
interactors:
  UsuarioCadastrado:
    term: human user
    description: Os Usuário acessam as funcionalidades na interface
Web.
processes:
  PRODUTOWEB:
    term: application server
    description: aplicacao web
```

Continuação na próxima página

Código UTML (continuação)

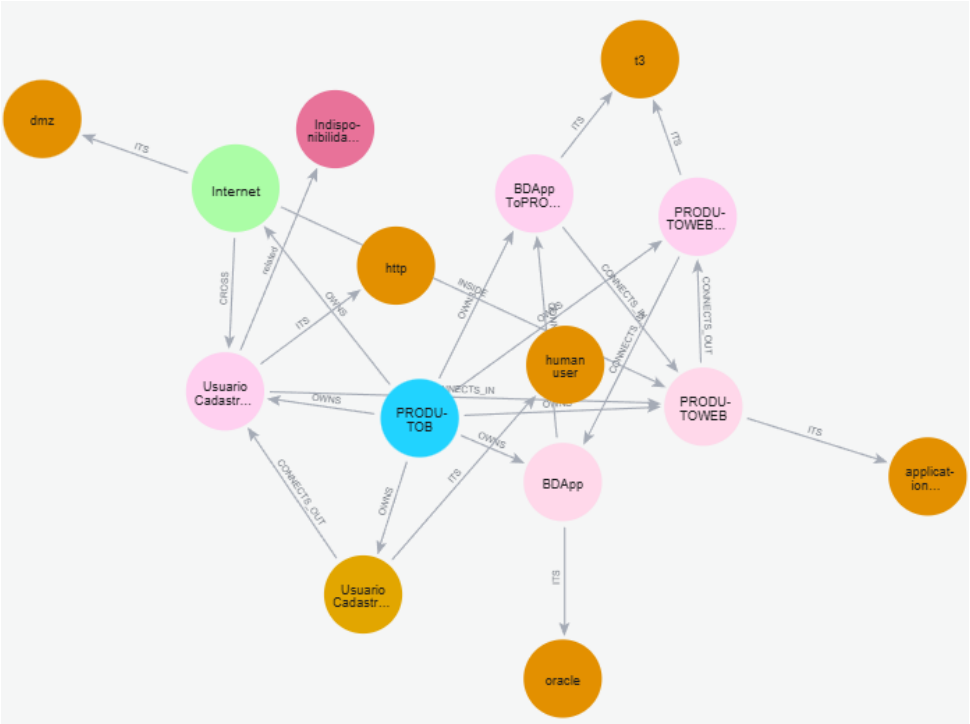
```
stores:
  BDApp:
    term: oracle
    description: Banco com todas informacoes
dataflows:
  UsuarioCadastradoToPRODUTOWEB:
    source_uid: br.produtos.UsuarioCadastrado
    destination_uid: br.produtos.PRODUTOWEB
    description: Request
    term: http
  PRODUTOWEBToBDApp:
    source_uid: br.produtos.PRODUTOWEB
    destination_uid: br.produtos.BDApp
    description: Request
    term: t3
  BDAppToPRODUTOWEB:
    source_uid: br.produtos.BDApp
    destination_uid: br.produtos.PRODUTOWEB
    description: "Dados e retorno de respostas"
    term: t3
trustboundaries:
  Internet:
    term: dmz
    description: Zona de Rede Internet
    crosses:
      - dataflow_uid: br.produtos.UsuarioCadastradoToPRODUTOWEB
    inside:
      - element_uid: br.produtos.PRODUTOWEB
threatflows:
  - element_uid: br.produtos.UsuarioCadastradoToPRODUTOWEB
threats:
  - threat_uid: DTP-1
```

Fonte: O Autor.

A Tabela 12 descreve o modelo de ameaça do sistema (*PRODUTOB*), incluindo seus principais componentes: o ator externo (*UsuárioCadastrado*) com o termo (*human user*), que utiliza a aplicação (*PRODUTOWEB*) rotulada como (*application server*) via fluxo de dados com o termo (*http*), conectada ao banco de dados (*BDApp*) com o termo (*oracle*) por meio de fluxo de dados com o termo (*t3*). A zona de confiança (*Internet*), rotulada pelo termo (*dmz*), delimita o acesso entre o usuário e o servidor. O modelo especifica ainda uma ameaça (DTP-1) associada ao fluxo entre o usuário e a aplicação, permitindo identificar potenciais riscos de segurança no tráfego de dados do sistema.

O algoritmo de conversão realiza a leitura do arquivo descrito em linguagem YAML e transforma suas entidades em uma estrutura de grafo. Nesse processo, cada elemento do modelo, como processos, armazenamentos de dados, fluxos, limites de confiança e atores, é representado como um nó, enquanto suas interações são registradas como relacionamentos navegáveis. Outros nós relacionados a termos e ameaças também são associados aos nós dos elementos. A partir dessa representação, o sistema torna-se capaz de executar as operações do mecanismo de recomendação, viabilizando consultas que permitem a seleção, decomposição, comparação e posterior recomendação de ameaças com base nos nós e conexões do grafo. A Figura 15 exibe a versão do grafo armazenada no banco de dados orientado a grafos.

Figura 15 – Resultado da conversão de um DFD na UTML para um Grafo



Fonte: O Autor

Na Figura 15, os termos (*dmz, http, human user, t3 e oracle*) estão destacados na cor laranja escura, indicando a sua função semântica no nó. Os elementos de processo (*PRODUTOWEB*), armazenamento (*BDApp*) e ator (*UsuarioCadastrado*) estão representados na cor rosa-claro, enquanto os fluxos de dados (*UsuarioCadastradoToPRODUTOWEB, PRODUTOWEBToBDApp e BDAppToPRODUTOWEB*) são exibidos em rosa. O limite de confiança (*Internet*) é apresentado em verde claro, e a ameaça associada ao fluxo de dados (DTP-1) aparece em rosa escuro. A diferenciação de cores dos nós do grafo tem por objetivo evidenciar o agrupamento dos elementos conforme seu tipo e papel dentro do modelo de ameaças.

3.1.4 Linguagem de descrição dos vocabulários de termos

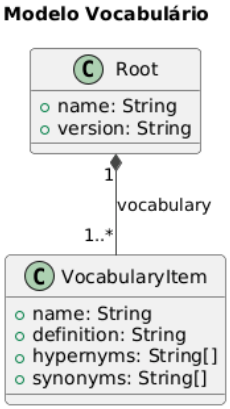
Os termos que rotulam os elementos e fluxos de um diagrama de fluxo de dados definem a sua característica principal. Eles são similares ao conceito de tags de tecnologia apresentado em Ramos, Silva e Castro (2018), em que representam etiquetas que identificam as tecnologias necessárias para o desenvolvimento de um requisito de software no contexto de um sistema de recomendação para requisitos

não funcionais. Neste trabalho, os termos são utilizados para auxiliar na determinação da similaridade dos elementos e fluxos que compõem um modelo de ameaças.

A representação das etiquetas possíveis em um modelo de ameaças é feita através da definição de um vocabulário comum para um conjunto de termos presentes nos modelos de ameaça avaliados. Para cada termo é possível definir uma relação hierárquica de hiperonímia, hiponímia e sinonímia com outros termos, funcionando como uma WordNet (MILLER, 1995) de propósito específico para modelagem de ameaças. A hiperonímia e seu oposto, a hiponímia, representam, respectivamente, uma relação de generalização e especialização entre conceitos. Por exemplo, o termo *application server* pode ser considerado um hipônimo de *server*. Já a sinonímia representa a definição de sinônimos semânticos, permitindo comparações mesmo entre termos diferentes.

O uso deste vocabulário permitiu calcular a similaridade entre termos sintaticamente diferentes e semanticamente parecidos. Para descrever o vocabulário e suas relações de hierarquia, foi especificado um modelo de descrição em YAML convertível para o formato WN-LMF (BOND et al., 2020). Este formato de descrição de uma Wordnet é interpretado pelo framework em Python WN (GOODMAN; BOND, 2021), que é o utilizado para permitir o cálculo de similaridade entre termos, por exemplo, através de algoritmos como o Wu e Palmer (1994) e Leacock e Chodorow (1998). A Figura 16 apresenta um exemplo da linguagem de descrição de WordNet simplificada.

Figura 16 – Linguagem de descrição dos Vocabulários



Fonte: O Autor

Na Figura 16 são definidos os seguintes tipos de dados para representar o vocabulário:

- Root: Define as informações básicas como nome e versão.
- VocabularyItem: Descreve o nome, a definição do termo, quais os hiperônimos, ou seja, os termos mais genéricos em uma árvore, e quais os sinônimos, ou seja, os termos equivalentes.

Para validar a linguagem de descrição do vocabulário, foi utilizado como exemplo um vocabulário mínimo que utilizasse todos os tipos de dados e, ao final, foi feita a conversão para a linguagem WN-LMF. A Tabela 13 apresenta o resultado desta conversão.

Tabela 13 – Descrição da taxonomia de termos em YAML

Vocabulário em YAML

```
name: Global DFD Vocabulary
version: 2

vocabulary:
  - name: device
    definition: Objeto físico capaz de executar hypernyms
  - name: input device
    definition: Dispositivo utilizado para entrada de dados
    hypernyms: [device]
  - name: keyboard
    definition: Dispositivo de entrada com teclas
    hypernyms: [input device]
  - name: smartphone
    definition: Dispositivo portátil
    hypernyms: [computing device, input device]
    synonyms: [cellphone]
```

Fonte: Autor.

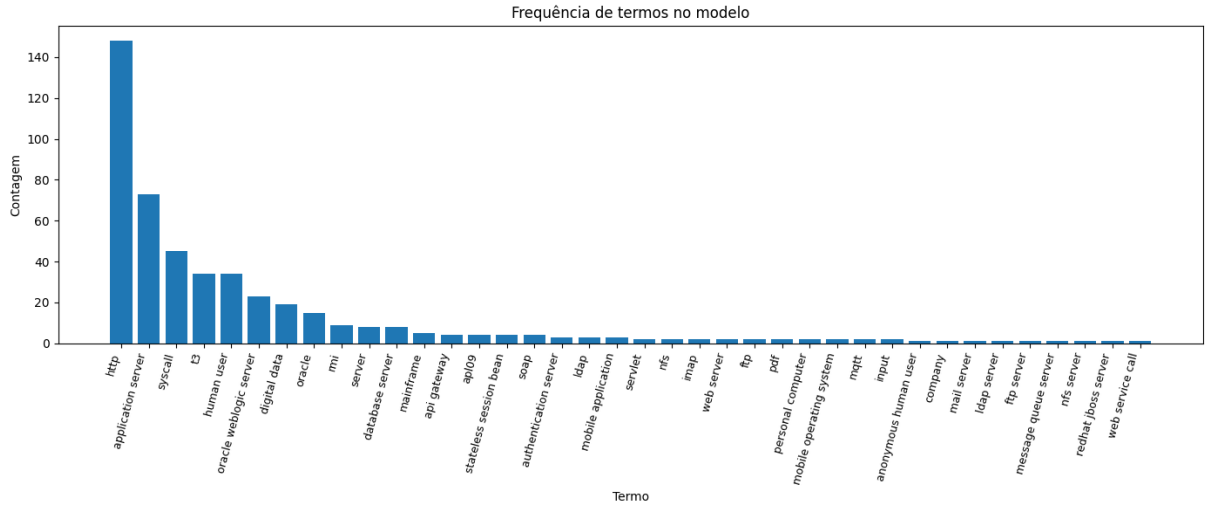
No exemplo da Tabela 13, enquanto a versão demonstrada possui 625 caracteres, a versão em XML da WN-LMF necessita de 3331 caracteres para representar a mesma estrutura hierárquica de uma WordNet. A decisão tomada nesta pesquisa de desenvolver uma linguagem de descrição de propósito específico teve como objetivo simplificar o processo de manipulação e gestão dos termos.

Para avaliar a presença dos termos nos modelos de ameaças do conjunto de dados, foi realizada uma consulta que contabilizou o número de termos utilizados pela organização. A Figura 17 mostra o número de vezes em que o termo foi utilizado.

De acordo com a Figura 17 observa-se uma predominância significativa de termos associados a aplicações web, como *application server, http e oracle weblogic server*, destacando que a maior parte dos sistemas modelados segue uma arquitetura baseada em aplicações web corporativas. Esse padrão também evidencia a presença recorrente de elementos voltados à autenticação, serviços de negócio e comunicação entre componentes distribuídos, como *stateless session bean, t3 e authentication server*. Termos como *mobile application, syscall, api gateway e soap* indicam diversidade tecnológica, contemplando desde aplicações móveis até sistemas que dependem de integrações baseadas em APIs. Por outro lado, elementos como *mainframe, nfs e ftp* aparecem com menor frequência, sugerindo que tecnologias legadas e serviços de suporte não são tão representativos no conjunto de modelos analisados.

Para facilitar a compreensão das relações conceituais, foi criada uma representação visual da WordNet que organiza e interliga os termos identificados no conjunto de dados deste trabalho. Ressalta-se que termos adicionais foram criados para uso nos modelos futuros. Essa representação pode ser consultada no Apêndice B.

Figura 17 – Frequência dos termos nos modelos de ameaça



Fonte: O Autor

3.1.5 Avaliação do conjunto de dados de modelos de ameaça

Os 34 modelos selecionados foram convertidos para a linguagem UTML e as etiquetas dos componentes dos diagramas de fluxo de dados, que não foram previstas nos modelos de ameaças, foram definidas para cada componente e fluxo de dados.

Para a atividade de limpeza dos dados, os modelos com diagramas de fluxo de dados com pequenos erros foram corrigidos. Na agregação dos dados, a lista de ameaças foi correlacionada entre diferentes descrições de forma a ter uma única lista de ameaças padronizada para todos os modelos, resultando ao final em 34 ameaças. Todas as ameaças tiveram suas descrições melhoradas com o objetivo de ter sua definição mais clara e concisa. A Tabela 14 abaixo apresenta um exemplo de melhoria realizada na descrição da ameaça.

Tabela 14 – Exemplo de melhoria na descrição das ameaças

Situação	**ID**	**Descrição**
Antes	DTP-8	Desabilitação de controles de segurança.
Depois	DTP-8	Comprometimento da proteção do sistema causado por desabilitação de controles de segurança.

Fonte: O Autor.

No exemplo da Tabela 14, a ameaça foi descrita de forma resumida, informando apenas a causa da ameaça, sem mencionar o impacto. Este processo buscou deixar mais claras as descrições das ameaças. Ao término do processo de refinamento das descrições, foi consolidada uma lista unificada de ameaças que serviu como base de conhecimento para o presente trabalho. Essa lista resultou da etapa de agregação e padronização das ameaças identificadas nos diferentes modelos analisados. A Tabela 15 apresenta o conjunto final de ameaças obtidas após o processo de consolidação. A descrição dos controles de segurança para cada ameaça pode ser vista no Apêndice C.

Tabela 15 – Agregação de todas as ameaças identificadas

ID	**Ameaça**
DTP-1	Indisponibilidade devido a esgotamento de recurso (processamento, largura de banda, armazenamento).
DTP-2	Exposição de informações causada por ausência de confidencialidade na transmissão de dados.
DTP-3	Alteração não autorizada de informações causada por falha ou ausência de integridade na transmissão de dados.
DTP-4	Acesso indevido causado por falha ou ausência de controle de autenticação de usuários ou clientes.
DTP-5	Acesso não autorizado a funcionalidades ou dados causado por falha ou ausência de controle de autorização para usuários autenticados.
DTP-6	Impossibilidade de rastrear eventos de segurança causada por ausência ou baixa qualidade de logs de auditoria.
DTP-7	Processamento de dados inválidos causado por ausência de validação adequada em dados de entrada.
DTP-8	Comprometimento da proteção do sistema causado por desabilitação de controles de segurança.
DTP-9	Exposição a ataques de aplicações web causada por ausência de mecanismos ativos de filtragem e proteção no tráfego HTTP.
DTP-10	Exposição de informações causada por violação de confidencialidade na entrega de tokens JWT.
DTP-11	Exfiltração de informações sensíveis causada por envio não autorizado para serviços de nuvem externa.
DTP-12	Exposição de segredos sensíveis causada por empacotamento indevido junto com a aplicação.
DTP-13	Alteração ou criação fraudulenta de documento causada por falsificação de documento digital.
DTP-14	Inconsistência de informações causada por falha na sincronização de dados.
DTP-15	Exposição de informações de infraestrutura causada por coleta de dados de identificação (*fingerprinting*).
DTP-16	Execução de código malicioso causada por entrada de arquivos não confiáveis para processamento.
DTP-17*	Processamento incompleto de tarefas causado por execução parcial de aplicação *batch*.
DTP-18	Extração automatizada de informações causada por coleta não autorizada de dados (*scraping*).
DTP-19	Aceitação de informações falsificadas causada por ausência de autenticidade na transmissão de dados.
DTP-20	Acesso indevido causado por falta de restrição a serviços internos.

ID	Ameaça
DTP-21	Acesso indevido causado por falsificação do servidor de controle de acesso.
DTP-22	Exposição de credenciais causada por interceptação da credencial de acesso do usuário.
DTP-23	Acesso não autorizado a recursos ou informações.
DTP-24	Exposição de credenciais causada por ausência ou uso de criptografia insegura no armazenamento de senhas.
DTP-25	Indisponibilidade de serviço causada pelo uso de expressões regulares de alto custo de processamento (*ReDoS*).
DTP-26*	Perda definitiva de informações causada por ausência de *backup*.
DTP-27	Engano do usuário causado por falsificação de site.
DTP-28	Exposição a vulnerabilidades no navegador causada por ausência de regras de segurança nos *headers* HTTP.
DTP-29	Intercepção, alteração ou vazamento de dados durante a transferência de arquivos (B2B) causada pelo uso de canais de comunicação inseguros ou não autenticados.
DTP-30	Intercepção, falsificação ou acesso indevido à comunicação entre produtos.
DTP-31	Vazamento, alteração ou perda de integridade de arquivos transmitidos entre sistemas internos causada pelo uso de mecanismos não padronizados ou inseguros de transferência.
DTP-32*	Acesso não autorizado, movimentação lateral ou propagação de ataques entre ativos internos causada por ausência de segmentação de rede e controle granular de comunicações.
DTP-33	Modificação maliciosa do conteúdo de um token de credencial (JWT) para acesso não autorizado.
DTP-34	Acesso não autorizado por ausência ou falha na verificação do token de credencial de acesso propagada para o produto interno.

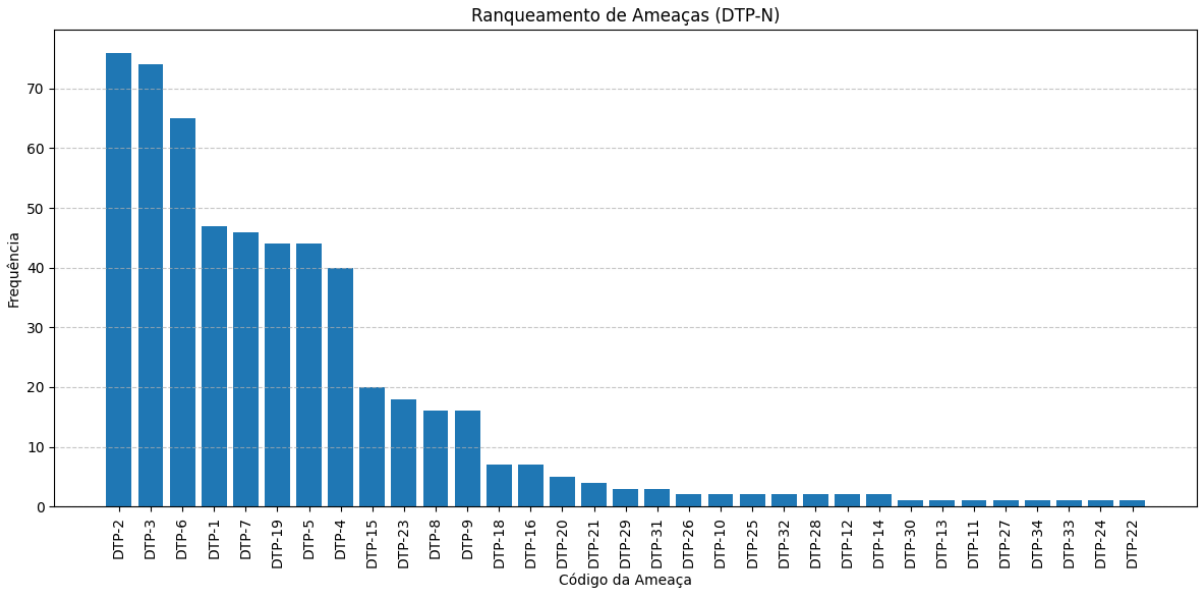
Fonte: O Autor.

A ameaça DTP-1 é referente a negação de serviço causada pelo esgotamento de um recurso, seja ele memória, processamento ou disco. Um exemplo real da concretização dessa ameaça é quando um sistema fica fora do ar por excesso de usuários simultâneos. A ameaça DTP-2 é referente a interceptação de informações por ausência de confidencialidade, sendo muito comum, quando um sistema não tem seu canal de comunicação criptografado. Outra ameaça, desta vez mais específica, é a DTP-34 que se refere a falta de verificação ou falha quando um token que representa uma credencial é propagado das camadas superiores para as camadas inferiores de um sistema.

As ameaças DTP-17, DTP-26 e DTP-32 foram desconsideradas por não estarem relacionadas ao design do sistema, mas sim a processos de infraestrutura. Assim, após sua exclusão, o ranqueamento das ocorrências das ameaças no conjunto de unidades de ameaça é apresentado na Figura 18.

As ameaças DTP-2 e DTP-3 são as mais indicadas e estão relacionadas com o controle de criptografia de canal. São seguidas pelas ameaças relativas ao controle de logs adequados e de proteção

Figura 18 – Ranqueamento de ameaças por unidade de ameaças

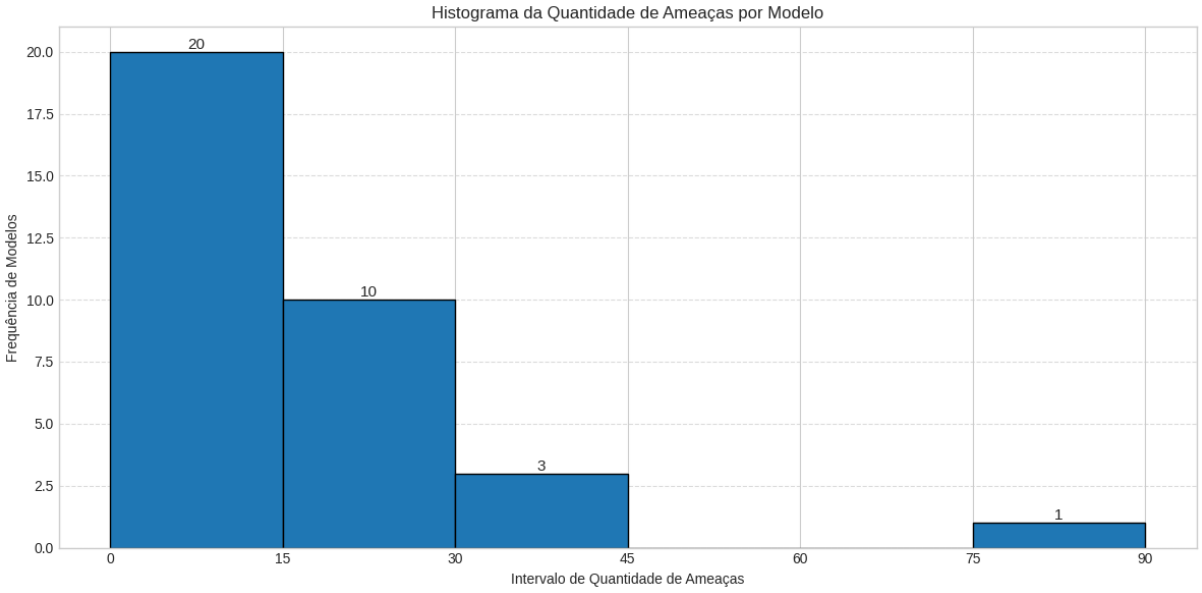


Fonte: O Autor

contra negação de serviço. Esta análise indica uma preocupação dos modelos de ameaças em garantir a integridade, confidencialidade e autenticidade das solicitações aos sistemas, com a rastreabilidade das atividades e a garantia da disponibilidade do sistema.

Para avaliar o conjunto de dados quanto à quantidade de ameaças identificadas por modelo, utilizou-se um histograma que ilustra a distribuição de frequências observada entre os modelos, conforme apresentado na Figura 19.

Figura 19 – Histograma de Quantidade de Ameaças por Modelo



Fonte: O Autor

Na Figura 19, observou-se uma concentração significativa nas faixas inferiores, em que 59%

dos modelos apresentaram até 15 ameaças. À medida que o número de ameaças por modelo aumenta, observa-se uma redução gradual na frequência, sugerindo que modelos de ameaças considerando uma ampla quantidade de ameaças são menos recorrentes no contexto analisado. Essa distribuição, com tendência à concentração nos intervalos iniciais, sugere que a atividade de modelagem de ameaças, na prática organizacional observada, identifica apenas um conjunto de poucas ameaças.

3.2 Arquitetura do sistema de recomendação

Filho (2021), sugeriu que a decomposição de uma arquitetura de um sistema de recomendação para engenharia de software em três componentes principais: o coletor, que é responsável por buscar as informações referente ao item pesquisado na base de conhecimento; o transformador, que é responsável por transformar o item pesquisado em um vetor de características que descreve o item; por fim, o recomendador, que é responsável por analisar as características, comparar a similaridade dos itens e recomendar os itens mais similares.

Esta pesquisa adotou uma estrutura baseada no modelo conceitual apresentado por Filho (2021), adaptando-a ao domínio de um sistema de recomendação de ameaças, da seguinte forma:

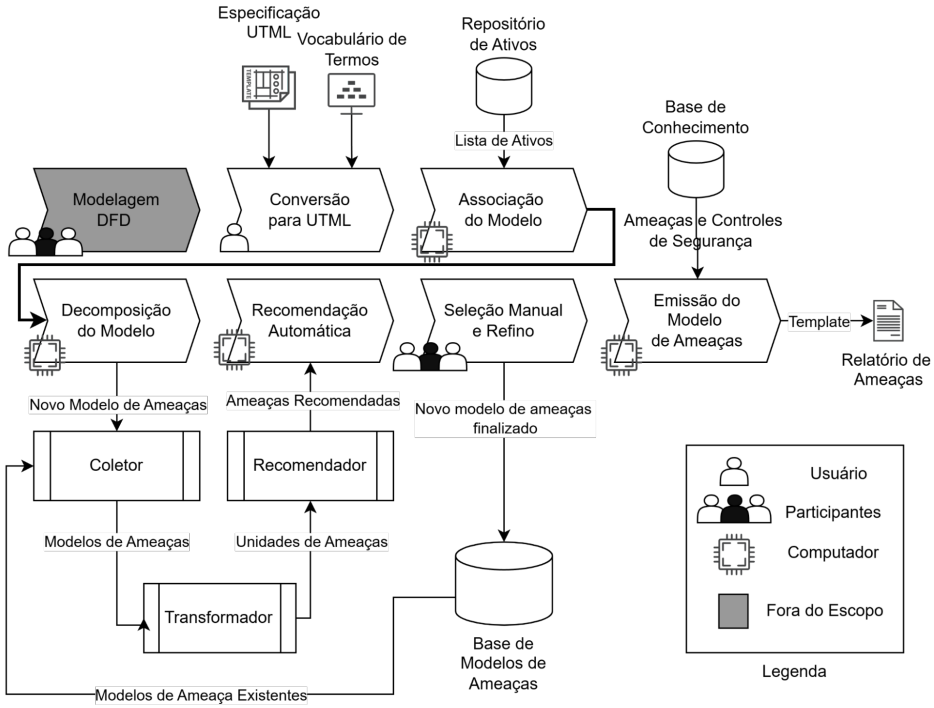
- **Coletor:** Responsável por obter os elementos e fluxos do Diagrama de Fluxo de Dados (DFD) do sistema sob análise, a partir do arquivo UTML. Também coleta da base de conhecimento as unidades de ameaças já registradas em modelos anteriores, armazenadas no banco de dados de grafos.
- **Transformador:** Executa a extração e padronização das características semânticas dos termos que compõem cada fluxo do DFD, gerando representações comparáveis para os fluxos, permitindo o cálculo de similaridade.
- **Recomendador:** Utiliza as características transformadas para calcular o grau de similaridade entre os fluxos do modelo atual e as unidades de ameaças existentes na base de conhecimento, empregando métricas de similaridade baseadas em vocabulário e estrutura de grafo. A partir desse ranqueamento, recomenda as ameaças mais prováveis para cada fluxo do novo modelo, compondo a lista de ameaças sugeridas.

A Figura 20 apresenta de forma integrada as atividades e componentes do sistema de recomendação proposto nesta pesquisa.

Para a utilização da ferramenta, são necessárias sete atividades para a finalização de um modelo. Entre estas atividades, seis representam atividades cobertas pelo sistema *Threat Copilot* e uma, que é a atividade inicial de modelagem do DFD, não foi escopo desta pesquisa. Abaixo segue a descrição de cada uma destas atividades.

- **Modelagem do DFD:** A modelagem do DFD ocorre nas etapas iniciais do processo de desenvolvimento seguro, durante a fase de design da arquitetura do sistema. Trata-se de uma atividade colaborativa, conduzida pelo time de desenvolvimento em conjunto com arquitetos e especialistas

Figura 20 – Atividades e Componentes do sistema de recomendação



Fonte: O Autor

em segurança da informação, utilizando os artefatos técnicos disponíveis sobre o sistema em construção. Seu objetivo é representar de forma estruturada os elementos, fluxos de dados e limites de confiança relevantes para a análise de segurança. No contexto desta pesquisa, o DFD já consolidado é utilizado como ponto de partida para o mecanismo de recomendação. Os aspectos relacionados à condução da sessão de modelagem, ao esforço necessário para sua realização e à qualidade do modelo produzido não compõem o escopo deste estudo, uma vez que o foco está na etapa subsequente de elicitação automática de ameaças a partir de um modelo já definido.

- **Conversão para UTML:** Essa atividade consiste em transformar o DFD, originalmente representado de forma gráfica, em uma descrição textual estruturada conforme a especificação da linguagem UTML. Durante esse processo, cada elemento do diagrama (atores, processos, armazenamentos e fluxos de dados) recebe um conjunto de termos (etiquetas semânticas) contidos num vocabulário de termos pré-definidos que padronizam sua representação e permitem o cálculo de similaridade no mecanismo de recomendação. O resultado dessa etapa é um arquivo em formato YAML, que se torna a entrada para as etapas subsequentes

- **Associação do modelo de ameaças:** consiste em realizar o relacionamento entre os componentes de um diagrama de fluxo de dados com a base de dados de ativos presente na organização através de um identificador único global, desta forma, permitindo a rastreabilidade das ameaças com seus ativos.

- Decomposição do modelo de ameaças: consiste em automaticamente navegar no grafo do diagrama de fluxo de dados, decompor seus componentes e relacionamentos em um conjunto de caracterís-

ticas denominado de unidade de ameaça. O conjunto de unidades de ameaça de um modelo de ameaças representa a consulta de entrada que será feita ao recomendador para que ele forneça as recomendações na atividade de recomendação automática.

- **Recomendação Automática:** é a apresentação do conjunto de recomendações de ameaça para todos os fluxos de dados do modelo de ameaças analisado com base nas decisões de recomendação do Recomendador.

- **Seleção manual e refino das ameaças:** Após a recomendação automática, o usuário realiza uma revisão crítica para validar a pertinência das ameaças sugeridas e complementar o modelo com novas ameaças que não tenham sido identificadas. Esse processo de curadoria resulta no modelo de ameaças finalizado, que é então armazenado na base de conhecimento da ferramenta. Assim, qualquer ameaça incluída manualmente passa a integrar o conjunto de ameaças disponíveis para análise de similaridade, possibilitando que seja recomendada automaticamente em avaliações futuras. Esse caráter incremental garante a evolução contínua da base de ameaças da organização, tornando o sistema progressivamente mais completo e eficaz ao longo do tempo.

- **Emissão do modelo de ameaças:** consiste em converter o modelo de ameaças finalizado em um artefato de relatório estruturado contendo as informações necessárias para subsidiar o tratamento das ameaças de segurança no sistema analisado. O documento resultante apresenta a lista de ameaças identificadas, sua categorização, a localização de cada ameaça no diagrama de fluxo de dados e os controles de segurança recomendados a partir da base de conhecimento da organização. Esse relatório torna-se o artefato formal de saída do processo, apoiando a comunicação com equipes de desenvolvimento de software e a tomada de decisões quanto à aplicação de controles de segurança.

3.2.1 O conceito de unidade de ameaça

Para representar as características identificadas pelo componente Transformador, foi estabelecido o conceito de unidade de ameaça de fluxo, que é o conjunto de atributos que caracterizam um fluxo de dados em um diagrama de fluxo de dados (DFD) de um modelo de ameaças. Essa definição é orientada a modelos de ameaças baseados em relacionamentos, correspondendo à abordagem de análise de ameaças por interação adotada pelo método STRIDE (SHOSTACK, 2014). A unidade de ameaça de fluxo é composta por três componentes essenciais:

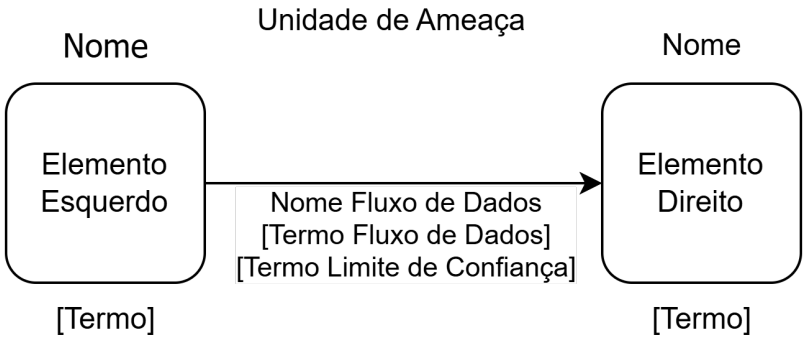
1. o elemento de origem do fluxo, localizado à esquerda, que representa o ponto de entrada das informações;

2. o fluxo de dados propriamente dito, responsável pela transmissão ou transformação da informação entre os elementos; e

3. o elemento de destino, posicionado à direita, que recebe, armazena ou processa os dados transmitidos.

A Figura 21 apresenta a notação gráfica utilizada para representar uma unidade de ameaça de fluxo, destacando as relações entre seus componentes e sua função na estrutura do modelo de ameaças e um exemplo de comparação de similaridade entre duas unidades de ameaça.

Figura 21 – Unidade de Ameaça



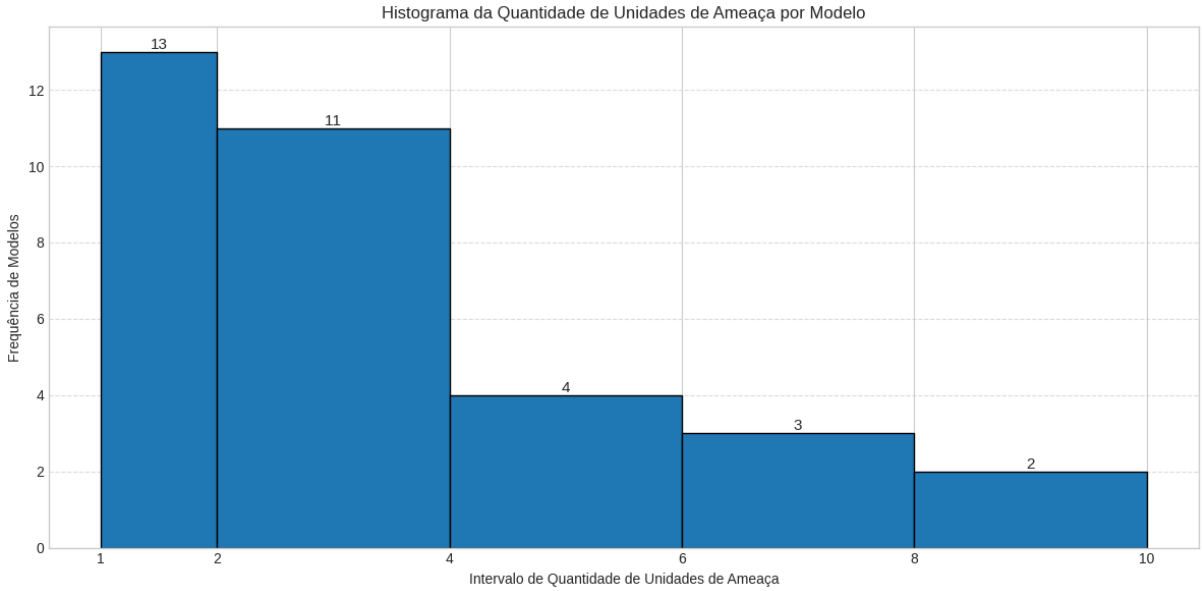
Fonte: O Autor

Os atributos de uma unidade de ameaça são:

- Nome do Elemento Esquerdo: Identifica o componente de origem do fluxo de dados no modelo de ameaças.
- Tipo do Elemento Esquerdo: Categoriza o elemento de origem (ex.: Ator Externo, Processo, Armazenamento de Dados).
- Termo do Elemento Esquerdo: Representa o termo do vocabulário associado ao elemento de origem.
- Nome do Fluxo de Dados: Corresponde ao nome ou descrição da informação transferida entre os elementos.
- Termo do Fluxo de Dados: É a definição do termo do vocabulário que define a característica principal do fluxo de dados, sendo comumente associado a protocolos de comunicação, como por exemplo, “http”.
- Termo do Limite de Confiança: Representa o termo do vocabulário que define a característica do limite de confiança que intercepta o fluxo de dados, quando existente.
- Nome do Elemento Direito: Corresponde ao nome do elemento de destino do fluxo de dados.
- Tipo do Elemento Direito: Categoriza o tipo do elemento de destino, seguindo o mesmo padrão do elemento esquerdo.
- Termo do Elemento Direito: Termo do vocabulário que define a característica principal do elemento de destino. Sua função é sintetizar a função essencial exercida pelo elemento no modelo. Por exemplo, caso o elemento do tipo processo fosse um servidor de aplicação, o termo escolhido seria “application server”.

Com o objetivo de compreender a distribuição da complexidade dos modelos de ameaças presentes no conjunto de dados utilizado, foi analisada a quantidade de unidades de ameaças associadas a cada modelo armazenado. Essa análise permitiu observar padrões relevantes sobre o tamanho e a amplitude dos modelos já construídos pela organização. A Figura 22 apresenta um histograma que ilustra essa distribuição, evidenciando o número de modelos de ameaça para cada faixa de quantidade de unidades de ameaça identificadas.

Figura 22 – Histograma da Quantidade de Unidades de Ameaça por Modelo



Fonte: O Autor

Observando a Figura 22, percebeu-se que a maior parte dos modelos concentra-se em até quatro unidades de ameaças por modelo, indicando que as atividades de modelagem de ameaças realizadas até o momento detalharam cenários de sistemas de menor complexidade arquitetural.

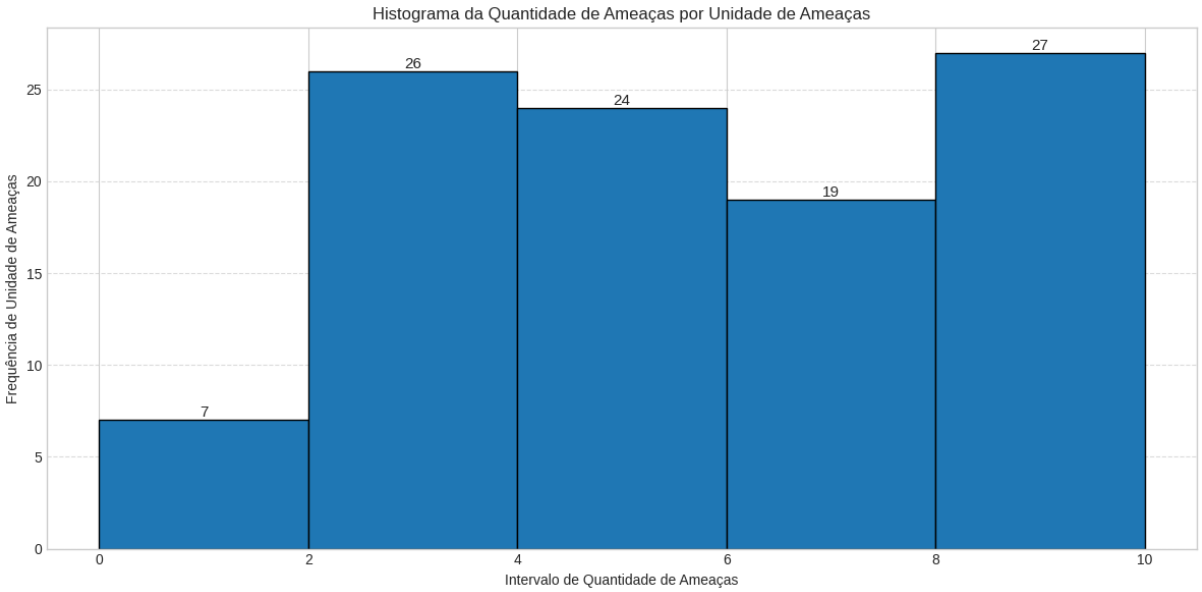
Para analisar a distribuição das unidades de ameaças identificadas nos modelos, elaborou-se um histograma que representa a frequência com que diferentes quantidades de ameaças ocorrem nas unidades de ameaça do conjunto de dados. A Figura 23 permite observar a concentração dos dados e identificar possíveis padrões de recorrência entre os valores registrados.

Observa-se que, com exceção do intervalo de 0 a 2 ameaças, as demais faixas apresentam uma distribuição relativamente equilibrada entre si. Essa uniformidade indica que cerca de 23% das unidades de ameaça concentram-se em cada uma das faixas analisadas, evidenciando uma distribuição proporcional das ameaças entre os diferentes níveis de ocorrência.

3.3 Heurística para Seleção das Unidades de Ameaças Similares

De acordo com (AGGARWAL, 2016), sistemas de recomendação baseados em conteúdo buscam identificar a relação de usuários com seus itens de forma similar ao que foi escolhido no passado. Esta similaridade não é necessariamente baseada na correlação de avaliações dos usuários, mas na base dos

Figura 23 – Histograma Unidade de Ameaças por Quantidade de Ameaças



Fonte: O Autor

atributos dos objetos utilizados e avaliados pelo usuário. Nesta pesquisa, o usuário é representado pelas novas unidades de ameaça geradas durante a modelagem, enquanto os itens disponíveis para recomendação correspondem às unidades de ameaça já cadastradas no banco de dados.

Sendo assim, adaptando a definição genérica do problema de recomendação para sistemas de recomendação baseados em conteúdo definido por (ADOMAVICIUS; TUZHILIN, 2005), temos:

- Seja UA_{new} o conjunto de todas as unidades de ameaça decompostas a partir de um novo modelo de ameaças;
- Seja UA_{history} o conjunto de todas as unidades de ameaça pré-existentes no banco de dados;
- Seja $\mathbb{R}$ um conjunto totalmente ordenado de números reais que representa o escore de utilidade;
- Seja $u : UA_{\text{new}} \times UA_{\text{history}} \rightarrow \mathbb{R}$;
- Seja $ua_{\text{n}} \in UA_{\text{new}}$ and $ua_{\text{h}} \in UA_{\text{history}}$.

O objetivo do sistema de recomendação é, para cada nova unidade de ameaça ua_{n} pertencente às unidades de ameaça do novo modelo de ameaça $(\forall, ua_n \in UA_{\text{new}})$, encontrar uma unidade de ameaça histórica ua'h que maximiza $(\arg\max ua_h \in UA_{\text{history}})$ a função de similaridade $u(ua_{\text{n}}, ua_{\text{h}})$. A Equação 1 apresenta a representação do problema de recomendação para o contexto desta pesquisa.

$$\forall ua_n \in UA_{\text{new}}, ; ua'_h = \arg \max_{ua_h \in UA_{\text{history}}} u(ua_{\text{n}}, ua_{\text{h}}) \tag{1}$$

A equação representa a ideia de que , para cada nova unidade de ameaça ua_n fornecida, o sistema de recomendação calcula o escore de similaridade baseado em conteúdo contra todas as unidades de

ameaça passadas $ua_h \in UA_{\text{history}}$. As unidades históricas ua_h que produzem os maiores escores são, então, escolhidas e as ameaças associadas são recomendadas. As próximas subseções visam definir os elementos que compõem o cálculo das unidades de ameaça mais similares para uma nova unidade de ameaça.

3.3.1 A similaridade entre nomes e tipos dos elementos de um DFD

Os nomes definidos para os elementos em sua maioria são nomes que indicam os componentes do sistema (exemplo: “SistemaABCD”, “SistemaBatch”, “BancoDeDadosAPP”, etc), sendo muitas vezes siglas ou acrônimos, não apresentando muito contexto semântico. Desta forma, optou-se por utilizar uma abordagem simplificada utilizando uma relação binária de igualdade entre duas cadeias de caracteres. Seu objetivo é determinar se dois nomes são idênticos, retornando o valor “1” (um) quando há correspondência exata e “0” (zero) caso contrário. Desta forma, permite que elementos sejam reutilizados em diversos modelos de ameaças, por exemplo, que um sistema de informação que fornece uma API com dados básicos para uma organização tenha precedência. Neste sentido, foi definida a função da Equação 2 para a comparação entre nomes de elementos de uma unidade de ameaças.

$$sim_{\text{nome}}(s_1, s_2) = \begin{cases} 1, & \text{se } s_1 = s_2 \\ 0, & \text{se não} \end{cases} \tag{2}$$

Onde:

- $s_1, s_2 \in \Sigma$, em que Σ corresponde a todos os caracteres possíveis.

A Equação 2 define uma função binária de similaridade nominal entre dois elementos, denotados por s_1 e s_2. A função $sim_{nome}(s_1, s_2)$ assume valor igual a 1 quando as cadeias de caracteres dos nomes comparados são idênticas, indicando correspondência exata, e 0 caso contrário. O conjunto N representa o universo de todos os caracteres possíveis, e o operador indica a relação de igualdade aplicada sobre os pares de caracteres.

Para a similaridade dos tipos dos elementos, utilizou-se a mesma abordagem adotada para o nome. Por exemplo, em uma comparação, caso o elemento do diagrama de fluxo de dados seja do tipo “Processo”, ele somente terá o valor de “1” (um) no caso de ser outro elemento do tipo “Processo”. Foi definida a função da Equação 3 para a comparação entre os tipos de elementos de uma unidade de ameaças.

$$sim_{\text{tipo}}(t_1, t_2) = \begin{cases} 1, & \text{se } t_1 = t_2 \\ 0, & \text{se não} \end{cases} \tag{3}$$

Onde:

- $t_1, t_2 \in T$, e T corresponde aos elementos possíveis em um diagrama de fluxo de dados (Ator, Armazenamento e Processo).

A Equação 3 estabelece a função de similaridade entre os tipos dos elementos que compõem uma unidade de ameaça. Essa função compara dois tipos t_1 e t_2 pertencentes ao conjunto T, que inclui as

categorias fundamentais do diagrama de fluxo de dados: Ator Externo, Processo e Armazenamento. O resultado da função $sim_{tipo}(t_1, t_2)$ é binário, assumindo o valor 1 quando ambos os elementos pertencem ao mesmo tipo e 0 caso contrário.

Essa abordagem objetiva garantir que elementos equivalentes tenham uma maior pontuação.

3.3.2 A similaridade entre termos

A rotulação dos elementos, fluxos de dados e limites de confiança de um diagrama de fluxo de dados de um modelo de ameaças representa o elemento central na identificação da similaridade de uma unidade de ameaças. Neste trabalho, o termo é definido como a característica principal que descreve um elemento, fluxo ou limite de confiança. Nos modelos de ameaças, os termos podem variar, por exemplo, "*usuário*", "*http*", "*banco de dados*" e "*servidor web*". O conjunto de termos pertencente a uma unidade de ameaças de fluxo descreve, portanto, a composição semântica dessa unidade, refletindo a relação entre o elemento de origem, o fluxo e o elemento de destino.

A partir dessa representação, foi possível comparar diferentes unidades de ameaça com base nas semelhanças entre seus termos, identificando correspondências conceituais entre fluxos de dados de distintos modelos. Assim, a similaridade entre termos é definida como uma função que mede o grau de equivalência entre os termos que compõem cada unidade. Diferente da similaridade de nome e tipo, a similaridade de termos neste trabalho considerou as relações hierárquicas de hipônimos e hiperônimos, e de equivalência por sinônimos em um vocabulário de palavras, similar às abordagens definidas por Gomes (2004) e Elkamel, Gzara e Ben-Abdallhah (2016).

O vocabulário de uma língua é um conjunto W de pares (f, s), em que uma forma (f) é uma sequência de caracteres sobre um alfabeto finito, e um sentido (s) é um elemento de um determinado conjunto de significados. As formas podem ser enunciados compostos por uma sequência de fonemas ou inscrições compostas por uma sequência de caracteres. Cada forma com um sentido em uma linguagem é definida como uma palavra (MILLER, 1995).

Em sistemas de informação, a representação de um vocabulário pode ser feita através de um WordNet. A forma é representada por uma cadeia de caracteres ASCII e o sentido é representado por uma ou um grupo de palavras que possuem o mesmo sentido (MILLER, 1995). Atualmente, existem diversos WordNets disponíveis online em diversas línguas. Os trabalhos de Gomes (2004) e Elkamel, Gzara e Ben-Abdallhah (2016) utilizaram uma Wordnet de língua inglesa para a recomendação de classes UML similares. Diferentemente destas abordagens e considerando o contexto da modelagem de ameaças, muitos dos termos utilizados não são comuns aos vocabulários gerais. Neste sentido, este trabalho optou por utilizar um vocabulário de propósito específico para a modelagem de ameaças no formato de uma WordNet, permitindo comparações semânticas entre termos distintos.

De acordo com Banu, Fatima e Khan (2013) existem quatro métodos para comparação de termos em uma WordNet: comparação baseada em contagens de arestas; comparação baseada em conteúdo de informação; comparação baseada em funcionalidades; métodos híbridos. Como este trabalho optou por ter sua própria WordNet de propósito específico, foi possível considerar apenas a categoria de comparação de contagens de arestas, já que informações essenciais para as demais categorias não foram definidas.

O método de contagens de arestas considera o vocabulário como um grafo que conecta de forma hierárquica a taxonomia dos termos, e a contagem das arestas entre dois termos é a medida de de similaridade entre eles, ou seja, quanto maior a distância no grafo entre os termos, menos similares eles são (CHANDRASEKARAN; MAGO, 2021). De acordo com Banu, Fatima e Khan (2013) nesta categoria os principais algoritmos de comparação são:

- Rada et al. (1989): a similaridade é inversamente proporcional ao tamanho do menor caminho entre dois termos. Neste método, a profundidade da hierarquia não influencia no cálculo da similaridade. A Equação (4) define o cálculo do algoritmo de menor distância.

$$sim_{\text{shortest-path}}(t_1, t_2) = \frac{1}{1 + length(t_1, t_2)} \tag{4}$$

  Onde:

  - $\text{length}(t_1, t_2) =$ número de arestas do caminho mais curto

  A Equação 4 define a similaridade entre dois termos t_1 e t_2 como o inverso da soma entre 1 e o comprimento do menor caminho que conecta esses termos na hierarquia semântica. O parâmetro $length(t_1, t_2)$ representa o número de arestas do caminho mais curto entre dois nós. O resultado é um valor entre 0 e 1.

- Wu e Palmer (1994): se baseia na profundidade hierárquica dos termos e do seu ancestral comum mais próximo (Least Common Subsumer – LCS). A similaridade aumenta à medida que o ancestral comum está mais próximo dos termos comparados.

$$sim_{\text{wup}}(t_1, t_2) = 2 \times \frac{\text{depth}(\text{LCS})}{\text{depth}(t_1) + \text{depth}(t_2)} \tag{5}$$

  Onde:

  - $\text{depth}(\text{LCS})$ corresponde à profundidade do ancestral comum mais baixo dos termos t_1 e t_2;
  - $\text{depth}(t_n)$ representa a profundidade do termo t_n na hierarquia conceitual.

  A Equação 5 define que a similaridade entre dois termos t_1 e t_2 é obtida a partir da relação entre a profundidade do seu menor ancestral comum (LCS) e as profundidades individuais desses termos na hierarquia. O termo depth(LCS) representa o nível hierárquico em que se encontra o conceito mais específico que generaliza t_1 e t_2, contado a partir da raiz da hierarquia. Já $depth(t_1)$ e $depth(t_2)$ indicam as profundidades dos próprios termos. O resultado é um valor entre 0 e 1, no qual se aproxima de 1 quando os termos compartilham um LCS e estão em níveis de profundidade semelhantes e se aproxima de 0 quando o LCS é muito genérico, próximo a raiz.

- Leacock e Chodorow (1998): a similaridade entre dois termos é calculada a partir do menor número de arestas que os separa na hierarquia, normalizado pela profundidade máxima da taxonomia do vocabulário. Dessa forma, quanto menor for a distância entre os termos e quanto mais profunda for a estrutura taxonômica do vocabulário, maior será a similaridade resultante.

$$sim_{\text{lch}}(t_1, t_2) = -\log\left(\frac{length(t_1, t_2)}{2 \times D}\right) \tag{6}$$

Onde:

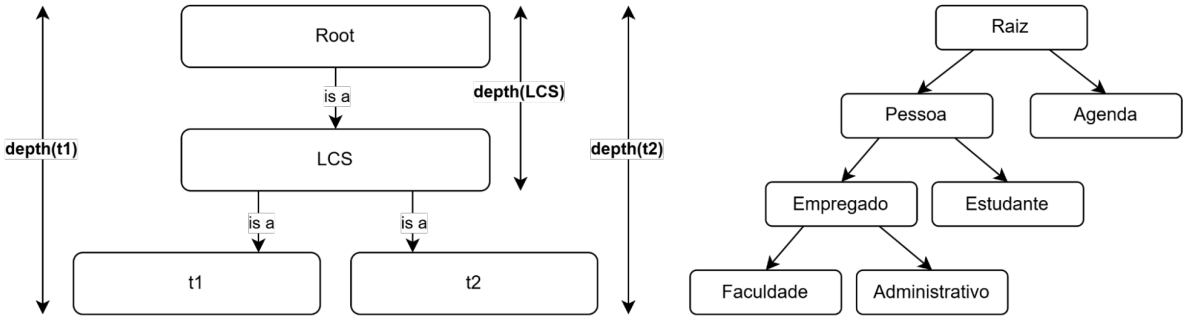
- $length(t_1, t_2)$ corresponde ao número de arestas do caminho mais curto entre os termos t_1 e t_2;
- D representa a profundidade máxima de todo o vocabulário.

A Equação 6 define que a similaridade entre dois termos t_1 e t_2 é obtida aplicando-se o logaritmo negativo da razão entre o comprimento do caminho $length(t_1, t_2)$ e o dobro da profundidade total do vocabulário.

Para o cálculo da similaridade semântica, optou-se pelo algoritmo de Wu e Palmer (1994). A principal motivação para esta escolha é que sua fórmula considera não apenas o ancestral comum mais baixo (LCS) entre dois conceitos, mas também a profundidade de cada conceito individualmente na taxonomia. Isso garante uma normalização eficaz, permitindo comparações justas entre termos de diferentes níveis de especificidade e tornando o algoritmo adequado para taxonomias parcialmente desbalanceadas, onde medidas baseadas apenas na distância de caminho poderiam gerar resultados distorcidos. A representação da árvore de hierarquia da WordNet pode ser visualizada no Apêndice B.

Para exemplificar o cálculo utilizando Wu e Palmer (1994) na Figura 24 foi definida uma hierarquia de exemplo. Neste exemplo, para calcular a similaridade entre o termo "Faculdade" e o termo "Administrativo", a profundidade de cada termo é "4" (quatro) e o ancestral mais comum (LCS) é "3" (três) ("Empregado"), o valor da similaridade $sim(t_1, t_2) = 2 \times 3 \div 4 + 4 = 0.75$. Para a similaridade entre os termos "Estudante" e "Faculdade", a profundidade para o termo "Estudante" é "3" (três), para o termo "Faculdade" é "4" (quatro), o ancestral mais comum (LCS) é "2" (dois), o valor da similaridade $sim(t_1, t_2)$ $= 2 \times 2 \div 3 + 4 = 0.57$.

Figura 24 – Exemplo de Funcionamento do Wu e Palmer (1994)



Fonte: O Autor

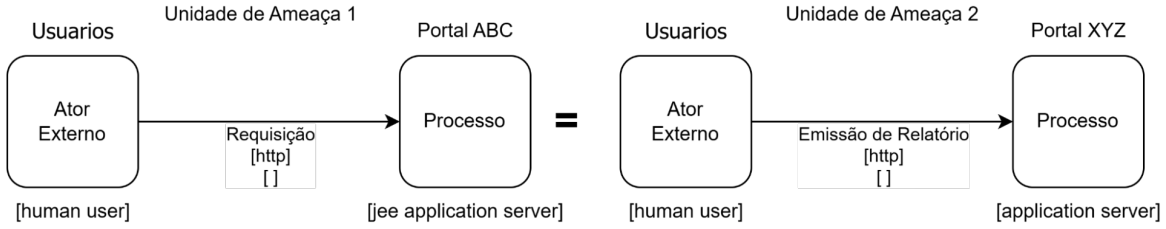
Este trabalho definiu sua função de similaridade para termos que rotulam os elementos de um DFD de um modelo de ameaças conforme a Equação 7.

$$sim_{\text{termo}}(t_1, t_2) = sim_{\text{wup}}(t_1, t_2) \tag{7}$$

3.3.3 A similaridade entre unidades de ameaças por pesos

A comparação entre duas unidades de ameaça se dá pela avaliação da similaridade dos atributos de uma unidade de ameaça utilizando as funções de similaridade por nome, por tipo e por termo. A Figura 25 apresenta um exemplo de atributos definidos para comparação.

Figura 25 – Comparação entre Unidades de Ameaças



Fonte: O Autor

No exemplo na Figura 25, temos os seguintes atributos avaliados quanto a similaridade entre duas unidades de ameaça: nome do elemento esquerdo (“Usuários” e “Usuários”), tipo do elemento esquerdo (“Ator Externo”, “Ator Externo”) , termo do elemento esquerdo (“human user” e “human user”), nome do fluxo (“Requisição” e “Emissão de Relatório”), termo do fluxo (“http”, “http”), termo do limite de confiança do fluxo (“” e “”), nome do elemento da direita (“Portal ABC”, “Portal XYZ”), tipo do elemento da direita (“Processo” e “Processo”) e termo do elemento da direita (“jee application server” e “application server”).

Para que cada atributo tivesse uma relevância específica na determinação da seleção da unidade de ameaça, foi definido que cada um dos atributos comparados teria um peso específico. Neste sentido, a Equação 11 representa o cálculo de similaridade baseada em pesos.

$$sim_{\text{fluxo}}(u_1,u_2) = \frac{p_1 \times sim_{\text{nome}}(u_1^{\text{fluxo}},u_2^{\text{fluxo}}) + p_2 \times sim_{\text{termo}}(u_1^{\text{fluxo}},u_2^{\text{fluxo}}) + p_3 \times sim_{\text{termo}}(u_1^{\text{lim}},u_2^{\text{lim}})}{\Sigma p} \tag{8}$$

$$sim_{\text{esq}}(u_1,u_2) = \frac{p_4 \times sim_{\text{tipo}}(u_1^{\text{esq}},u_2^{\text{esq}}) + p_5 \times sim_{\text{nome}}(u_1^{\text{esq}},u_2^{\text{esq}}) + p_6 \times sim_{\text{termo}}(u_1^{\text{esq}},u_2^{\text{esq}})}{\Sigma p} \tag{9}$$

$$sim_{\text{dir}}(u_1,u_2) = \frac{p_7 \times sim_{\text{tipo}}(u_1^{\text{dir}},u_2^{\text{dir}}) + p_8 \times sim_{\text{nome}}(u_1^{\text{dir}},u_2^{\text{dir}}) + p_9 \times sim_{\text{termo}}(u_1^{\text{dir}},u_2^{\text{dir}})}{\Sigma p} \tag{10}$$

$$sim(u_1,u_2) = sim_{\text{fluxo}}(u_1,u_2) + sim_{\text{esq}}(u_1,u_2) + sim_{\text{dir}}(u_1,u_2) \tag{11}$$

Em que:

- u_n representa uma unidade de ameaça;

- sim_{nome} corresponde à função de similaridade por nome;
- sim_{termo} corresponde à função de similaridade por termo de vocabulário;
- sim_{tipo} corresponde à função de similaridade por tipo de elemento;
- $p_{1...9}$ representam os pesos individualizados;
- Σp representa o somatório de todos os pesos.

Na Equação 11 a similaridade é entre duas unidades de ameaça u_1 e u_2 é medida pela ponderação da similaridade dos atributos dos fluxos de dados, do elemento esquerdo e do elemento direito das unidades de ameaças pelo peso de cada um destes atributos. O objetivo da utilização dos pesos é permitir que atributos específicos de uma unidade de ameaça sejam mais relevantes na seleção das unidades similares.

O método e os valores dos pesos adotados nesta função serão detalhados na próxima subseção, incluindo a justificativa técnica para sua definição, o processo empírico de consenso realizado com especialistas.

3.3.4 A definição dos pesos por consenso de especialistas

De acordo com (AGGARWAL, 2016), pesos podem ser utilizados para definir a importância das características para uma recomendação, a definição pode ser feita de forma automática ou utilizando o conhecimento de domínio. Diante da variabilidade e da pequena quantidade de amostras do conjunto de dados disponível, a abordagem de obter um conjunto de pesos para ponderar as características da unidade de ameaça de forma automática não seria adequada. Neste sentido, considerando um sistema de recomendação de propósito específico, optou-se por que o conhecimento de domínio seja utilizado na recomendação. A definição por experts também foi uma abordagem adotada pela ferramenta AutSEC (FRYDMAN et al., 2014) mas no contexto da definição dos custos associados aos controles de segurança para mitigação de riscos.

Este trabalho optou pelo método Delphi (LINSTONE; TUROFF, 1975) que é baseado no consenso de especialistas. De acordo com (MENDONÇA, 2022), o método Delphi é um processo estruturado e iterativo que solicita as opiniões de um grupo de especialistas em relação a um problema, tópico ou tarefa. Normalmente, os especialistas são submetidos a uma série de questionários, intercalados com informação condensada e feedback controlado. Ele já foi utilizado para determinar parâmetros críticos que afetam o e-Learning (SHIBANI et al., 2023), como também para definir pesos que influenciam um sistema de recomendação de trabalhos científicos baseado em reputação científica (MENDONÇA, 2022) e para estimar custos de projetos de software (GANDOMANI; WEI; BINHAMID, 2014).

Para Mendonça (2022), o método possui cinco características determinantes:

1. Os participantes são especialistas na sua área;
2. Os participantes assumem uma identidade anônima;
3. Os questionários são executados em iterações;

4. Os participantes recebem feedback ao longo do processo;

5. As respostas dos participantes devem ser estatisticamente agrupadas.

Com o objetivo de evitar uma sequência significativa de iterações, com impacto no tempo disponível dos participantes especialistas selecionados, e de orientar os especialistas na definição dos seus pesos iniciais com base nas percepções do grupo buscando a convergência via método Delphi utilizou-se previamente a técnica de Rank Order Centroid (ROC) (ROSZKOWSKA, 2013), que é utilizada para calcular os pesos iniciais a partir de um ranqueamento de prioridades.

No contexto deste trabalho, buscou-se determinar os pesos adequados para cada uma das características utilizadas na determinação da similaridade entre duas unidades de ameaças. Para esta definição foram executadas as seguintes etapas usando o método Delphi:

1. Inicialmente, foram selecionados especialistas da Organização X com experiência comprovada em modelagem de ameaças. Em seguida, realizou-se uma apresentação individual, por videoconferência, na qual foram explicados a motivação para a definição dos pesos, o conceito de unidade de ameaça e o mecanismo de comparação por similaridade.

2. Cada especialista foi convidado a estabelecer um ranqueamento das comparações de similaridade, ordenando-as de acordo com sua percepção de importância. Com base nesses ranqueamentos, foi aplicada a técnica *Rank Order Centroid* (ROC), que permitiu gerar automaticamente um conjunto inicial de pesos, o qual foi encaminhado aos especialistas para apoiar a etapa subsequente.

3. Para refinar os pesos com o método Delphi, conduziu-se uma rodada anônima de definição dos pesos, na qual cada participante, de posse das informações preliminares, atribuiu valores conforme sua experiência e julgamento técnico. Após essa rodada, cada especialista recebeu a média geral e o desvio padrão dos pesos atribuídos, podendo revisar suas estimativas a fim de buscar convergência.

4. Por fim, foi realizada uma rodada adicional de reuniões individuais, entre o pesquisador e o especialista para tratar especificamente dos critérios que apresentaram maior divergência. Nessas conversas, buscou-se esclarecer dúvidas e aprofundar a discussão sobre as propriedades em desacordo, de modo a apoiar a consolidação final dos pesos.

As atividades 3 e 4 se repetiram até o critério do desvio padrão ser ≤ 3 (em pontos percentuais). Os resultados foram agregados na média e no desvio padrão, e, quanto maior o desvio, maior a divergência. Neste tipo de atividade, espera-se que haja convergência após algumas rodadas.

Os especialistas selecionados foram analistas de segurança da informação pós-graduados da Organização X com experiência de mais de onze anos na área de desenvolvimento seguro e modelagem de ameaças. Eles estão dispersos nos estados da PB, RN e CE. A Tabela 16 apresenta as características dos especialistas.

Entre os participantes selecionados, dois possuem especialização e um possui mestrado. Todos possuem mais de 11 anos trabalhando com segurança da informação na área de desenvolvimento de software. A comunicação com cada um dos especialistas foi individual e online através da ferramenta de

Tabela 16 – Lista de Especialistas

Especialista	**Experiência em Segurança da Informação**	**Conhecimento**
Participante 1	14 anos	Especialização
Participante 2	11 anos	Especialização
Participante 3	13 anos	Mestrado

Fonte: O Autor.

comunicação Microsoft Teams (Microsoft Corporation, 2017). Inicialmente, foi solicitado para cada um dos participantes um ranqueamento sobre a relevância da similaridade de cada item relacionado com uma unidade de ameaça. A Tabela 17 indica a priorização dos pesos por cada participante.

Tabela 17 – Priorização dos Pesos

Peso	**Propriedade**	**P1**	**P2**	**P3**	**Média Ranking**
P1	sim_nome(fluxo)	9	4	7	6,67
P2	sim_termo(fluxo)	2	1	1	1,33
P3	sim_termo(boundary)	3	9	4	5,33
P4	sim_tipo(elemento_esquerdo)	6	7	9	7,33
P5	sim_nome(elemento_esquerdo)	8	5	6	6,33
P6	sim_termo(elemento_esquerdo)	4	2	3	3,00
P7	sim_tipo(elemento_direito)	5	8	8	7,00
P8	sim_nome(elemento_direito)	7	6	5	6,00
P9	sim_termo(elemento_direito)	1	3	2	2,00

Fonte: O Autor.

A Tabela 17 apresenta em suas colunas a identificação dos pesos, a propriedade que está sendo avaliada, a ordem de prioridade, sendo 1 para o mais importante, dada ao peso pelos participantes P1, P2 e P3, e por fim, a média do ranking. De acordo com a Tabela 17, pela ordenação da média das escolhas dos especialistas, o ranqueamento de relevância dos pesos seguiu a seguinte ordem crescente: P2, P9, P6, P3, P8, P5, P1, P7, P4.

A partir da ordem de prioridade dos pesos definida, foi possível determinar um conjunto de pesos iniciais. No ROC (ROSZKOWSKA, 2013) uma ordem de classificação é transformada em valores numéricos de peso, mantendo a proporcionalidade entre as posições. A Equação 12 apresenta a fórmula para calcular o peso de cada propriedade.

$$w_i = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{j} \tag{12}$$

Em que:

- n = número total de pesos
- i = posição da propriedade (1=mais importante)
- w_i peso da propriedade i.

A fórmula apresentada na Equação 12 A define o peso w_i como a média das inversas das posições a partir de j até n, garantindo que critérios mais bem ranqueados recebam maior peso. A normalização pelo total de critérios n assegura que a soma dos pesos seja igual a 1. A Tabela 18 apresenta os pesos iniciais calculados.

Tabela 18 – Proposta de Pesos Iniciais pelo Rank Order Centroid

Peso	P2	P9	P6	P3	P8	P5	P1	P7	P4
Valor	31,43	20,32	14,76	11,06	8,28	6,06	4,20	2,62	1,23

Fonte: O Autor.

De acordo com a Tabela 18, foi definido o peso de 31,43 para a propriedade mais importante (P2) que representa o termo que rotula o fluxo de dados e o peso de 1,23 para a propriedade menos importante (P4) que representa o tipo do elemento esquerdo.

Para refinar o conhecimento dos especialistas, foi definida uma tabela onde cada um dos especialistas atribuiu o valor a cada peso de acordo com o seu conhecimento e avaliação de importância. As três listas de pesos foram coletadas na primeira rodada e os resultados foram agrupados pela média, mediana, desvio padrão, mínimo, máximo e coeficiente de variação. A Tabela 19 apresenta os resultados obtidos após a primeira rodada.

Tabela 19 – Primeira Rodada de Definição de Pesos

P	**Propriedade**	**P1**	**P2**	**P3**	**Média**	**Mediana**	**Desvio**	**Min**	**Máx**	**Coef. Var (%)**
P1	sim_nome(fluxo)	5	13	7	8,33	7,00	4,16	5,00	13,0	49,96
P2	sim_termo(fluxo)	20	20	20	20,00	20,00	0,00	20,00	20,0	0,00
P3	sim_termo(limite)	15	3	15	11,00	15,00	6,93	3,00	15,0	62,98
P4	sim_tipo(el._esq.)	5	7	4,5	5,50	5,00	1,32	4,50	7,0	24,05
P5	sim_nome(el._esq.)	5	10	7	7,33	7,00	2,52	5,00	10,0	34,32
P6	sim_termo(el._esq.)	15	15	17,5	15,83	15,00	1,44	15,00	17,5	9,12
P7	sim_tipo(el._dir.)	10	7	4,5	7,17	7,00	2,75	4,50	10,0	38,42
P8	sim_nome(el._dir.)	5	10	7	7,33	7,00	2,52	5,00	10,0	34,32
P9	sim_termo(el._dir.)	20	15	17,5	17,50	17,50	2,50	15,00	20,0	14,29

Fonte: O Autor.

O critério adotado para avaliar se a próxima rodada deve ser realizada é que o desvio padrão seja inferior a 3. Analisando os resultados obtidos, nota-se que os pesos P1 e P3 não atenderam o critério, sendo necessária uma nova rodada. Uma observação é que os pesos P5, P7 e P8 embora atendam o critério, tiveram um coeficiente de variação significativo, demonstrando uma certa divergência entre os especialistas. As informações agregadas foram disponibilizadas aos participantes e uma nova rodada foi realizada. A Tabela 20 apresenta os resultados da segunda rodada.

Nesta segunda rodada os pesos P1 e P3 ainda não alcançaram o critério de parada. Houve uma pequena convergência mas não o suficiente, sendo necessária uma terceira rodada para a definição dos pesos. A Tabela 21 apresenta os resultados da terceira rodada.

Tabela 20 – Segunda Rodada de Definição de Pesos

P	Propriedade	P1	P2	P3	Média	Mediana	Desvio	Min	Máx	Coef. Var (%)
P1	sim_nome(fluxo)	5	13	9	9,00	9,00	4,00	5,00	13,0	44,44
P2	sim_termo(fluxo)	25	20	20	21,67	20,00	2,89	20,00	25,0	13,32
P3	sim_termo(limite)	10	3	12	8,33	10,00	4,73	3,00	12,0	56,71
P4	sim_tipo(el._esq.)	5	7	4,5	5,50	5,00	1,32	4,50	7,0	24,05
P5	sim_nome(el._esq.)	5	10	7	7,33	7,00	2,52	5,00	10,0	34,32
P6	sim_termo(el._esq.)	15	15	17,5	15,83	15,00	1,44	15,00	17,5	9,12
P7	sim_tipo(el._dir.)	10	7	4,5	7,17	7,00	2,75	4,50	10,0	38,42
P8	sim_nome(el._dir.)	5	10	8	7,67	8,00	2,52	5,00	10,0	32,83
P9	sim_termo(el._dir.)	20	15	17,5	17,50	17,50	2,50	15,00	20,0	14,29

Fonte: O Autor.

Tabela 21 – Terceira Rodada de Definição de Pesos

P	Propriedade	P1	P2	P3	Média	Mediana	Desvio	Min	Máx	Coef. Var (%)
P1	sim_nome(fluxo)	7,5	12	8,5	9,33	8,50	2,36	7,50	12,0	25,32
P2	sim_termo(fluxo)	22,5	20	20	20,83	20,00	1,44	20,00	22,5	6,93
P3	sim_termo(limite)	10	10	12	10,67	10,00	1,15	10,00	12,0	10,83
P4	sim_tipo(el._esq.)	5	5	4,5	4,83	5,00	0,29	4,50	5,0	5,97
P5	sim_nome(el._esq.)	5	9	7	7,00	7,00	2,00	5,00	9,0	28,57
P6	sim_termo(el._esq.)	15	15	17,5	15,83	15,00	1,44	15,00	17,5	9,12
P7	sim_tipo(el._dir.)	10	5	5	6,67	5,00	2,89	5,00	10,0	43,30
P8	sim_nome(el._dir.)	5	9	8	7,33	8,00	2,08	5,00	9,0	28,39
P9	sim_termo(el._dir.)	20	15	17,5	17,50	17,50	2,50	15,00	20,0	14,29

Fonte: O Autor.

Na terceira rodada o critério de parada foi alcançado. Os pesos P1 e P3 alcançaram o desvio padrão inferior a 3. Os pesos finais foram obtidos a partir da média desta última rodada e são apresentados na Tabela 22.

Tabela 22 – Pesos definidos pelos Especialistas

Peso	P2	P9	P6	P3	P8	P5	P1	P7	P4
Valor	20,83	17,50	15,83	10,67	7,33	7,00	9,33	6,67	4,83

Fonte: O Autor.

Diferente da priorização inicial, após as rodadas, o peso P1 que define a relevância da similaridade do nome utilizado no fluxo de dados teve seu ranqueamento da sétima posição para a quinta posição. Com estes pesos definidos com base no conhecimento dos especialistas, foi possível determinar a função final de similaridade entre unidades de ameaça.

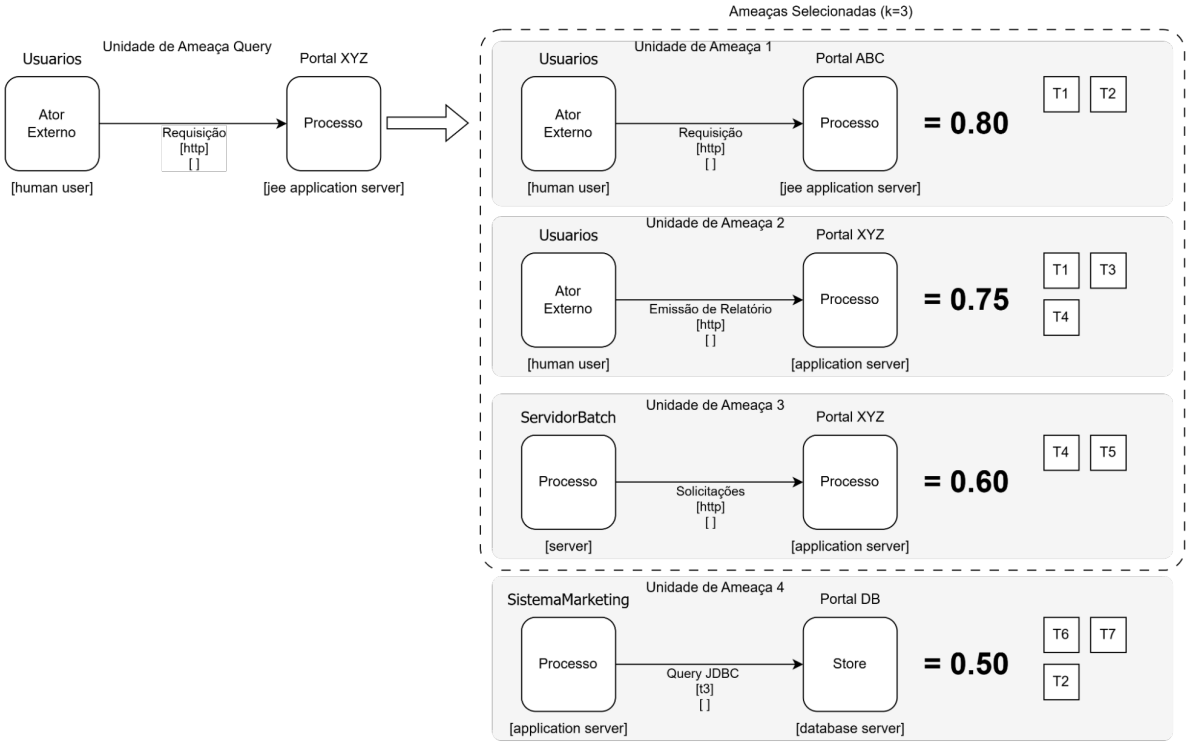
3.3.5 A Seleção das Ameaças Mais Similares

A seleção das ameaças mais similares utiliza a abordagem conhecida como ranqueamento por top-N. Ela é eficiente quando existem poucas unidades de análise, permitindo recomendações com base

na proximidade da consulta (ROBILLARD et al., 2014). No top-N, o algoritmo recebe como entrada um item para ser consultado, o algoritmo calcula a similaridade deste item com os itens armazenados na base de dados e, em seguida, define os *n* melhor ranqueados, ou seja, os itens mais similares ao consultado.

No contexto do sistema de recomendação proposto, uma unidade de ameaça de um novo modelo de ameaças é utilizada como uma consulta (*query*) ao sistema de recomendação. Ele aplica a função de similaridade da unidade de ameaças consultada com todas as unidades de ameaças presentes na base de conhecimento, ao final é feito um ranqueamento das unidades de ameaças mais similares, por fim, o conjunto das ameaças recomendadas é escolhido a partir do topo de n previamente definido. A Figura 26 detalha o funcionamento do método.

Figura 26 – Seleção das Ameaças (n=k)



Fonte: O Autor

No exemplo demonstrado na Figura 26, os escores de ranqueamento da similaridade são calculados individualmente entre a unidade de ameaça pesquisada e as unidades de ameaça da base de conhecimento. Considerando um n de 3 para identificar os vizinhos mais próximos, as Unidades de Ameaça 1, 2 e 3 são selecionadas com os valores de 0.8, 0.75 e 0.60 e as ameaças listadas para o solicitante serão as T1, T2, T3, T4 e T5.

Assim como relatado em Filho (2021), identificou-se a possibilidade de um comportamento não determinístico na seleção das ameaças quando ocorre o empate de similaridade entre duas unidades de ameaças. O algoritmo inicial escolhia aleatoriamente a unidade de ameaças baseada na disponibilização das unidades de ameaças do banco de dados orientado a grafos utilizados. Para obter um comportamento determinístico, optou-se por ordenar, de forma alfabética, a partir do nome completo de uma unidade de ameaça. O nome completo é composto pelo pacote de trabalho de uma organização e o nome do fluxo

de dados. Por exemplo, para uma "Organização X" e um fluxo de dados "EmissãoExtratoBancário", o nome completo seria "br.gov.orgx.modelo.emissaoextratobancario". Essa escolha prioriza as unidades de ameaças de uma mesma organização. Feita a ordenação, as primeiras unidades de ameaça mais similares são escolhidas para recomendação.

Selecionadas as ameaças, para representar a ordenação para exibição ao usuário, é utilizada uma função de escore que considera a quantidade de vezes que a ameaça foi recomendada e a severidade da ameaça. A Equação 13 representa esta função.

$$escore(t_i) = sim(t_i) \times \frac{severidade(t_i)}{severidade_{\max}} \times \frac{total(t_i)}{qtd_{\text{ameaças}}} \times 100 \tag{13}$$

Na Equação 13 o resultado da similaridade $sim(t_i)$ é ponderado pelo fator da severidade de ameaça, normalizado com o maior valor possível de severidade, e também pela frequência que a ameaça foi recomendada, igualmente normalizada em relação ao maior número máximo de recomendações possíveis.

Ressalta-se que a função de escore não interfere na seleção das ameaças recomendadas, atuando apenas na ordenação de sua exibição. Contudo, a ordenação pode introduzir um viés de apresentação, uma vez que ameaças mais frequentes e severas tendem a ocupar as primeiras posições da interface, o que pode influenciar a atenção e a escolha do analista. Sendo assim, reconhece-se como limitação que essa ordenação pode influenciar a decisão do usuário, favorecendo a escolha de ameaças mais frequentes e severas em detrimento de ameaças menos recorrentes, porém potencialmente relevantes.

3.4 A Ferramenta *Threat Copilot*

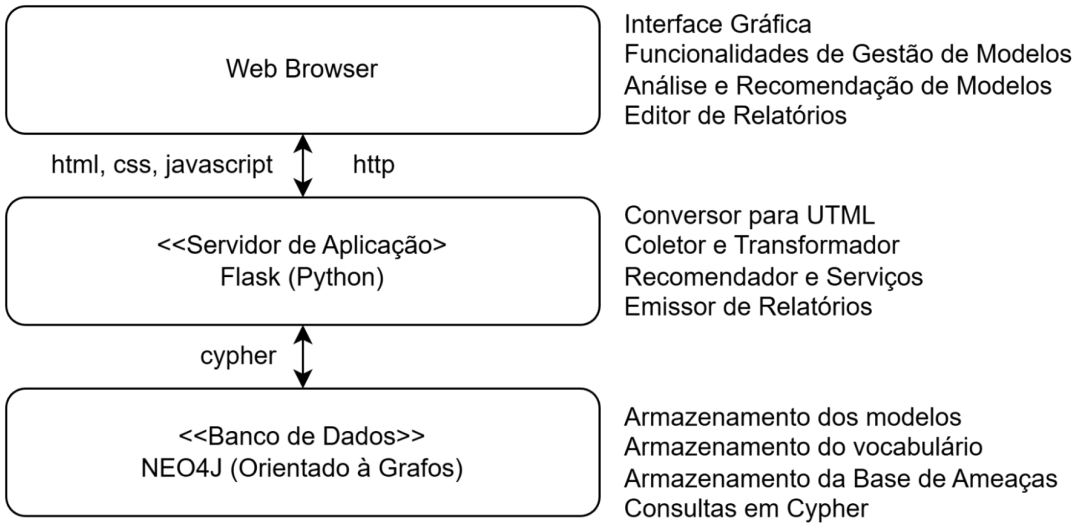
A ferramenta *Threat Copilot* foi implementada como uma aplicação web para apoiar a análise de viabilidade da solução, composta por uma arquitetura modular que integra *backend* e *frontend*.

O *backend* foi desenvolvido em Python (Python Software Foundation, 2025), utilizando o *framework Flask* (GRINBERG, 2018), que oferece simplicidade, baixo acoplamento e flexibilidade para a criação de APIs e serviços web. Optou-se por esta linguagem quando se considerou seu ecossistema de bibliotecas para ciência de dados, processamento linguístico e operações matemáticas. Neste cenário, o servidor de aplicação é responsável pelos componentes Coletor, Transformador e Recomendador, além de emitir relatórios e fornecer os serviços utilitários para os componentes.

No frontend, foram utilizadas tecnologias como HTML, CSS e JavaScript que estruturam a camada de apresentação da interface, garantindo interação intuitiva com o usuário durante a criação e inspeção dos modelos de ameaças. Para a camada de persistência e consulta do grafo que representa os modelos de ameaças, empregou-se a linguagem Cypher (FRANCIS et al., 2018) usando o banco de dados orientado a grafos Neo4J (Neo4j Inc., 2025), possibilitando consultas otimizadas sobre elementos e relações definidos nos diagramas de fluxo de dados. A Figura 27 apresenta a arquitetura básica da ferramenta.

Para apoiar o processamento e a comparação de termos utilizados nos modelos de ameaças, a ferramenta *Threat Copilot* utiliza a biblioteca WN - WordNet Framework (GOODMAN; BOND, 2021)

Figura 27 – Arquitetura Básica da Ferramenta *Threat Copilot*



Fonte: O Autor

como base léxico-semântica para o cálculo de similaridades entre termos, possibilitando a aplicação das métricas, como as medidas semânticas de similaridade baseadas em Wu e Palmer (1994).

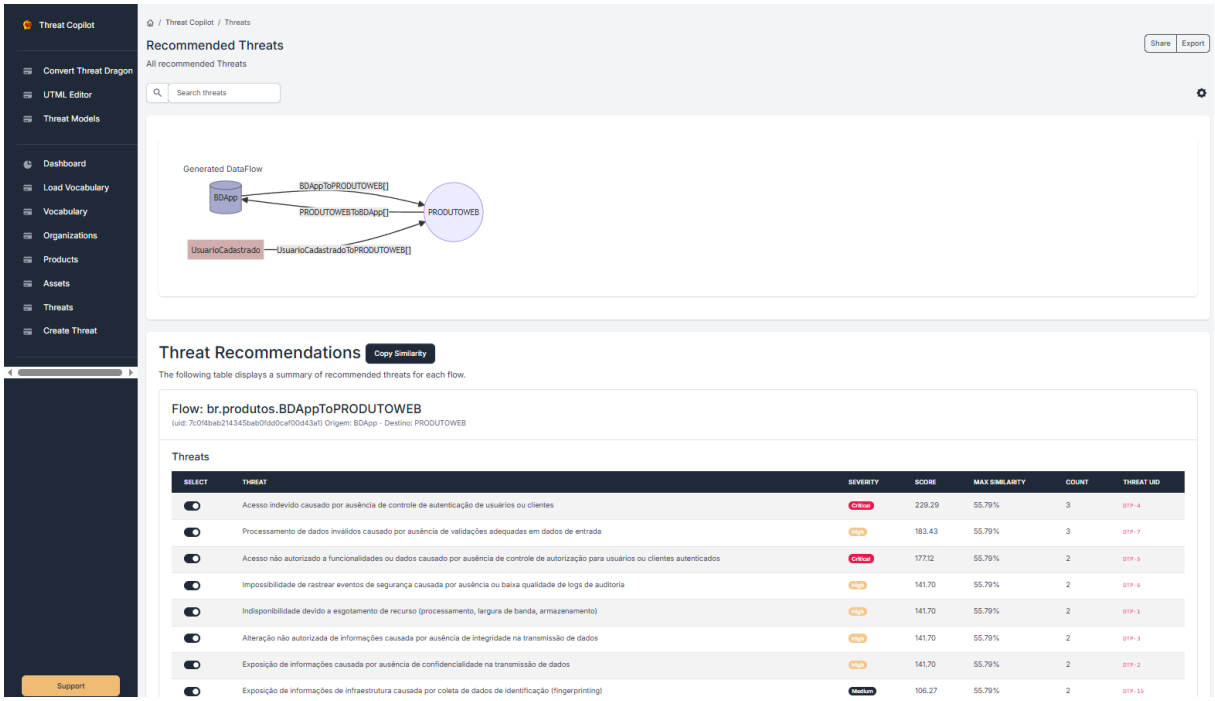
Para a persistência dos dados no banco orientado a grafos Neo4J, foi empregado o Neomodel (Neo Technology, 2025), um Object Graph Mapper (OGM) que estabelece uma camada de abstração entre os modelos de dados da aplicação e as operações em Cypher. O uso do Neomodel simplifica a definição dos nós e das relações correspondentes aos artefatos de modelagem de ameaças, garantindo consistência no armazenamento, facilidade de consulta e manutenção do dataset evolutivo da organização.

A visualização dos diagramas de fluxo de dados foi realizada por meio da biblioteca em javascript Mermaid (Mermaid, 2025), uma ferramenta baseada em marcação declarativa que permite a renderização automática do diagrama de fluxo de dados envolvido no modelo de ameaças. Além disso, a edição e formatação dos relatórios gerados no sistema utilizam a biblioteca em javascript Markdown-It (Markdown-it, 2025), que permite ao usuário construir documentação estruturada com maior flexibilidade e suporte a extensões para formatação avançada, mantendo a padronização dos artefatos produzidos pelo processo de modelagem de ameaças. A Figura 28 apresenta a tela de recomendação de ameaças da ferramenta com um exemplo fictício.

Neste exemplo, o DFD é gerado e exibido, apresentando ao usuário o desenho do modelo utilizado. Abaixo do DFD são apresentadas as recomendações de ameaças (*Threat Recommendations*) ordenadas pelo escore da Equação 14, com um campo selecionável (*Select*), no início de cada ameaça listada, que indica se a ameaça deve ser escolhida ou não para o fluxo observado. No Apêndice D são apresentadas as demais telas da ferramenta.

Atualmente, o *Threat Copilot* apresenta uma base de código composta por 5.426 linhas de código em Python, distribuídas em 43 arquivos, responsáveis principalmente pela lógica de negócio, pelos mecanismos de recomendação e pela integração com o banco de dados orientado a grafos. Complementarmente, a camada de interface apresenta 9.123 linhas de código HTML, organizadas em 60 arquivos, o que busca

Figura 28 – Tela de Recomendação da Ferramenta *Threat Copilot*



Fonte: O Autor

demonstrar o foco na construção de uma aplicação web interativa voltada ao suporte visual da modelagem de ameaças.

3.5 Decisões de design da ferramenta

No sistema de recomendação deste trabalho, denominado de *Threat Copilot*, os modelos de ameaças prévios definidos em dados não estruturados foram transformados em dados estruturados para que possam ser interpretáveis pelos algoritmos utilizados. As recomendações são geradas por meio de um modelo baseado em memória, no qual são selecionados, na base de conhecimento, os fluxos de dados mais similares àqueles presentes no modelo de ameaça em análise. Essa abordagem foi adotada em função da utilização do modelo STRIDE por interação pela Organização X avaliada. Por fim, aplicou-se o modelo de ranqueamento para ordenar as ameaças mais relevantes associadas a cada fluxo de dados.

A Tabela 23 descreve as características de design do sistema de recomendação adotado, considerando as principais funcionalidades definidas na Figura 6.

Em conjunto, essas funcionalidades sintetizam a lógica geral de operação do Threat Copilot, evidenciando a articulação entre estruturação dos dados, recuperação de conhecimento prévio e priorização das ameaças recomendadas.s

Tabela 23 – Decisões do Design da *Threat Copilot*

Fase	**Descrição**
Pré-processamento de Dados	*Parsing* de um grafo de um diagrama de fluxo de dados através de dados estruturados em YAML.
Captura de Contexto	Extração de palavras-chave e de nomes definidos no diagrama de fluxo de dados.
Produzir Recomendações	Algoritmos baseados em memória (conteúdo), considerando métricas de similaridade próprias, agrupadas com ranqueamento para a ordenação da listagem das ameaças.
Apresentar Recomendações	Interface Web que mostra os diagramas de fluxo de dados e apresenta as recomendações de ameaças específicas por fluxo.

Fonte: O autor.

4 AVALIAÇÃO DO SISTEMA DE RECOMENDAÇÃO

Neste capítulo discutimos os resultados da pesquisa. Ele foi dividido em três partes, iniciando pela avaliação das métricas *offline* da ferramenta. A segunda parte consiste na avaliação *online* centrada no usuário, onde são apresentados o protocolo, as questões, o estudo piloto, a caracterização do perfil dos participantes, a descrição dos construtos e suas variáveis, e a consolidação das respostas abertas. Por fim, são apresentadas as atividades, os resultados e as métricas obtidas durante a execução de uma sessão de modelagem *online* usando a ferramenta.

4.1 Avaliação *offline* da seleção das ameaças

Definido o método de seleção das ameaças (detalhado na seção 3.3.5), buscou-se avaliar o sistema de recomendação *Threat Copilot* para definir as métricas de desempenho da proposta. De acordo com García (2024), a avaliação de sistemas de recomendação pode ser realizada em dois momentos: a validação *offline*, realizada antes da disponibilização do sistema de recomendação, e a avaliação *online*, realizada após a implantação, e normalmente centrada no usuário.

A avaliação *offline* permite verificar as métricas de um sistema de recomendação utilizando um conjunto de dados conhecidos e isolados do mundo real. É mais fácil de conduzir, exige menos recursos e permite identificar as melhores decisões de design e algoritmos de um sistema de recomendação (RICCI; ROKACH; SHAPIRA, 2010). Entretanto, avaliações offline apenas representam dados do passado, não sendo possível prever o comportamento do sistema de recomendação quando em uso (FILHO, 2021).

Dentre as dimensões possíveis para avaliação de um sistema de recomendação para engenharia de software, no contexto da avaliação offline, este trabalho buscou determinar a dimensão da corretude para as ameaças mais interessantes no sistema de recomendação proposto (ROBILLARD et al., 2014). Para isto utilizou uma técnica bastante comum denominada de validação cruzada K-Fold (*K-Fold cross validation*), onde o objetivo é utilizar todo o conjunto de dados para teste e treinamento, garantindo que a métrica de desempenho adotada seja menos sensível a uma divisão de dados específica (ROBILLARD et al., 2014; FILHO, 2021).

Para a avaliação dos folds da técnica de validação cruzada, foram adotadas métricas *precision*, *recall* e *f1-measure*, que são comumente utilizadas em sistemas de recomendação aplicados à engenharia de software. Elas possuem o objetivo de mensurar a qualidade das listas de recomendação de ameaças (ROBILLARD et al., 2014; GARCÍA, 2024).

A *precision* é definida pelo percentual das recomendaçoes relevantes para o usuário em relaçao à quantidade de recomendaçoes retornadas (FILHO, 2021). Ou seja, entende-se a precisão como a métrica que avalia as ameaças válidas para uma unidade de ameaça. A precisão é definida através dos conceitos Positivo Verdadeiro (PV) e de Falsos Positivos (FP), na forma de uma atividade de classificação binária (GARCÍA, 2024). O PV indica o número correto de recomendações, enquanto o FP indica o número incorreto de recomendações. A Equação 14 apresenta o cálculo do *precision*.

$$precision = \frac{PV}{PV + FP} \tag{14}$$

A métrica *precision* da Equação 14 é definida pela razão entre o número de positivos verdadeiros (PV) e o total de itens recomendados pelo sistema, que inclui tanto os positivos verdadeiros e os falsos positivos (FP).

A métrica de *recall* é utilizada para avaliar o percentual de itens relevantes nas listas de recomendação, ou seja, das ameaças das que realmente deveriam ser identificadas e que foram corretamente recomendadas. O *recall* é derivado do conceito de PV e de Falso Negativo (FN). O FN indica a quantidade de itens não recomendados pelo sistema de recomendação que deveriam ser recomendados. A Equação 15 apresenta o cálculo do *recall*.

$$recall = \frac{PV}{PV + FN} \tag{15}$$

A métrica *recall* da Equação 15 é definida pela razão entre o número de positivos verdadeiros (PV) e o total de ameaças corretas, que é a soma entre os positivos verdadeiros (PV) e falsos negativos (FN).

A última métrica utilizada é denominada de *f1-measure*. Essa métrica fornece uma avaliação equilibrada do desempenho do sistema ao considerar o trade-off entre recomendar uma maior quantidade de ameaças, aumentando o *recall* porém com risco de falsos positivos, e recomendar apenas ameaças altamente confiáveis, aumentando o *precision* porém com risco de omitir ameaças relevantes. A Equação 16 apresenta o cálculo da *f1-measure*.

$$f1\text{-}measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{16}$$

Na Equação 16 a *f1-measure* é definida pela média harmônica entre as métricas de *precision* e *recall*. Essa métrica será apenas alta se tanto a *precision* quanto a *recall* forem elevadas.

No contexto da segurança da informação e da modelagem de ameaças, um *precision* baixo indica o custo de avaliar ameaças que não serão selecionadas, já um *recall* baixo representa o risco de que ameaças relevantes deixem de ser consideradas durante o processo de modelagem, naturalmente implicando na ausência de controles e maior exposição do sistema avaliado a riscos de segurança.

No contexto da avaliação *offline* deste trabalho, foi utilizado o K-Fold de 5, desta forma, o conjunto total de unidades de ameaças presente na base de dados (103) foi particionado em 5 subconjuntos de testes (20, 20, 21, 21 e 21) e 5 subconjuntos de treinamento (83, 83, 82, 82 e 82). Tal configuração assegura, em cada execução, aproximadamente 20% dos dados para teste e 80% para treinamento, seguindo a prática comum de particionamento. Para cada uma das cinco iterações do K-Fold foram avaliadas as métricas *precision*, *recall* e *f1-measure* considerando a média dos resultados de 1 a 5 os vizinhos mais similares (k). Os resultados obtidos podem ser observados na Tabela 24 abaixo.

Os resultados apresentados na Tabela 24 demonstraram o desempenho do sistema de recomendação de ameaças ao longo das cinco execuções de validação cruzada (*k-fold*). As métricas *precision*, *recall*

Tabela 24 – Resultados das Métricas de Avaliação Offline com Wu e Palmer (1994)

Fold	***precision***	***recall***	***f1-measure***
k-fold-1	66,88	76,37	68,89
k-fold-2	55,53	61,67	56,63
k-fold-3	54,18	80,24	61,67
k-fold-4	38,75	74,38	47,07
k-fold-5	39,33	66,68	47,20
Média	50,94	71,87	56,29

Fonte: O Autor.

e *f1-measure* foram definidas como colunas da tabela, estes valores são as médias dos resultados obtidos a partir da seleção do topo do ranqueamento de um a cinco unidades de ameaça. As linhas representam as iterações de um a cinco da divisão de folds de unidades de ameaças utilizados. A última linha representa a média geral das métricas para todos os folds. O fold 1 foi o que apresentou o melhor desempenho geral das métricas, com o maior valor de *precision*. O pior resultado foi identificado no fold 4 que apresentou o pior valor de *precision*, embora o *recall* seja superior a média. Nesse fold também temos o pior valor da métrica *f1-measure*.

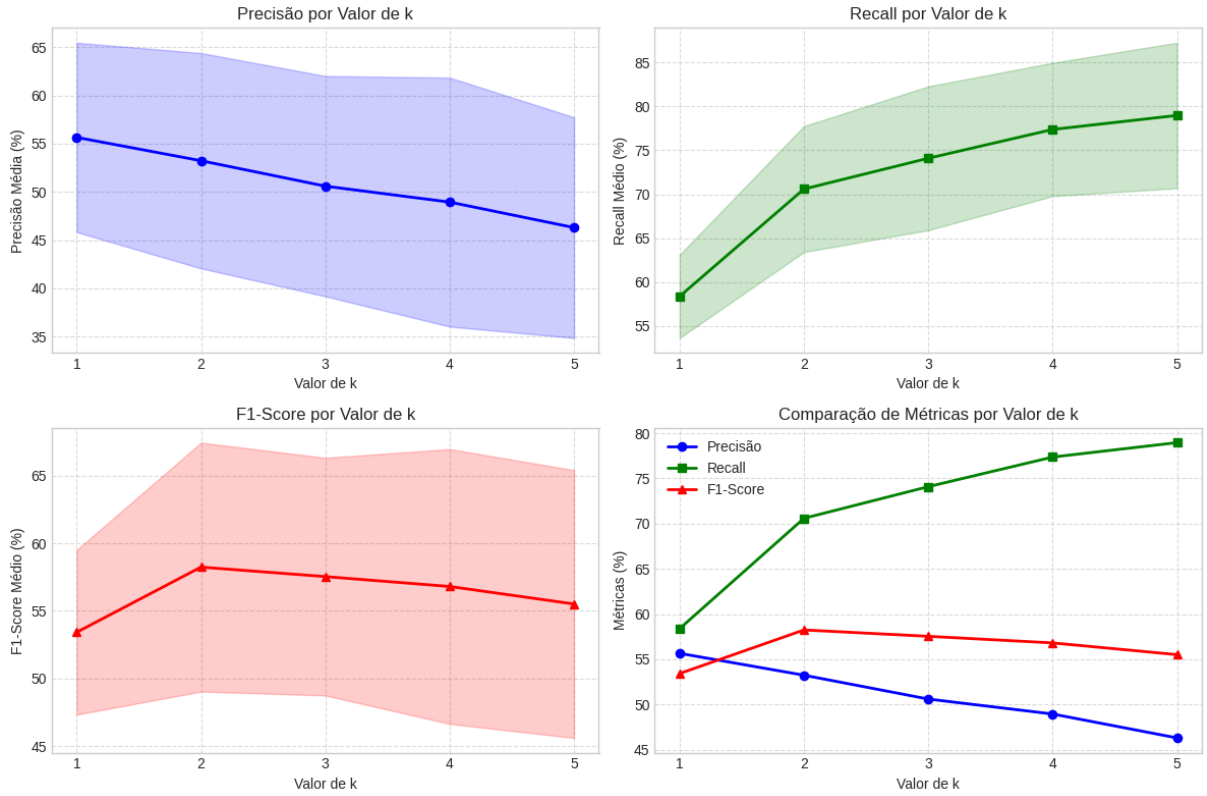
Observou-se que o *recall* apresentou valores superiores aos de *precision* em todos os folds, alcançando média de 71,87%, enquanto a média de *precision* foi de 50,94%. A *f1-measure*, que busca balancear essas duas métricas, apresentou desempenho intermediário, com média de 56,29%.

Considerando o mesmo teste sendo executado para os algoritmos comparativos Leacock e Chodorow (1998) temos os valores respectivos de *precision, recall e f1-measure*: para Rada et al. (1989) 51.05%, 71.83% e 56.52; e para Leacock e Chodorow (1998) 50.93%, 71.76%, 56.38%. Estes resultados sugerem que para o dataset e WordNet atual não existe diferenciação significativa nas métricas, sugerindo a limitação da quantidade de modelos e o uso de um vocabulário com uma hierarquia simplificada. A escolha pelo Wu e Palmer (1994) se dá por fundamentação teórica, não por superioridade empírica comprovada.

A segunda avaliação apresentada na Figura 29 mostra o comportamento do sistema de recomendação a conforme o valor de seleção do topo (k) aumenta, com o k variando de um a cinco. Observou-se uma tendência de queda gradual no *precision*, que passa de 55,65% para 46,30% entre k=1 e k=5. O cenário indica que ao ampliar a quantidade de ameaças consideradas na recomendação, aumentou também a ocorrência de falsos positivos. Por outro lado, o *recall* cresce de forma consistente, saindo de 58,38% para 78,96%, o que demonstra que valores maiores de k permitem recuperar um número maior de ameaças relevantes, reduzindo omissões de análise durante a modelagem de ameaças.

A determinação do k adequado para o ranqueamento top-N é realizada através das avaliações do conjunto das métricas, apesar de o valor de n=2 ter apresentado o maior *f1-measure* entre os cenários de *n* avaliados, optou-se por utilizar n=3 como configuração padrão. Essa decisão se justifica pelo fato de que existe um ganho percentual no *recall* importante (3.95%), sem perder tanto *precision* (2,64%). Neste sentido, considerando o contexto de segurança da informação, o *recall* é mais importante. O n=3 apresentou um equilíbrio com uma pequena vantagem no *recall* e, portanto, um cenário mais adequado entre esforço de análise e efetividade da recomendação, garantindo cobertura de 74,08% de ameaças relevantes ao mesmo tempo em que evita sobrecarregar o analista de segurança com sugestões inefetivas.

Figura 29 – Resultados das Métricas *Offline* por top-N (n=k) com Wu e Palmer (1994)



Fonte: O Autor

4.2 Avaliação *Online* Centrada no Usuário

Em sistemas de recomendação uma das formas de avaliar o seu funcionamento é através de estudos com usuários. Nele o avaliador seleciona diretamente os participantes e os convida a interagir com o sistema de recomendação para realizar tarefas específicas. O *feedback* pode ser coletado antes e depois da interação, e o sistema também pode registrar informações sobre o modo como o usuário interage com as recomendações (AGGARWAL, 2016).

Para avaliar a aceitação e adequabilidade da ferramenta de modelagem de ameaças automatizada, conforme previsto na Metodologia, foi utilizado o Modelo de Aceitação de Tecnologia (TAM) integrado ao Modelo de Ajuste Tecnologia Tarefa (TTF), fornecendo a base teórica necessária para permitir descrever as variáveis para identificar as perspectivas de uso da ferramenta proposta. Foram selecionadas as variáveis definidas por Cardoso e Cleophas (2020), e as questões foram adaptadas para o contexto desta pesquisa. A Tabela 25 apresenta as questões.

Para a condução do estudo e obtenção dos dados das variáveis apresentadas, adotou-se um procedimento estruturado em etapas sequenciais, conforme descrito a seguir:

1. Convite e seleção dos participantes com perfil aderente ao público-alvo da ferramenta, composto por profissionais com experiência em segurança da informação ou modelagem de ameaças.

2. Realização de um estudo piloto com objetivo de validar os questionários, a clareza dos itens

Tabela 25 – Questionário TAM-TTF

Construto	**ID**	**Variável**	**Descrição**
Utilidade Percebida	1	Desempenho	O uso da automação da modelagem de ameaças contribui para o desempenho das minhas tarefas?
	2	Produtividade	A automação da modelagem de ameaças melhora a produtividade da modelagem de ameaças?
	3	Rapidez	A automação da modelagem de ameaças possibilita a realização das tarefas mais rapidamente?
	4	Utilidade	A automação da modelagem de ameaças proposta é útil para a modelagem de ameaças?
Facilidade de Uso	5	Operação do Método	Considero fácil aprender a usar a automação da modelagem de ameaças?
	6	Simplicidade	Eu considero simples e intuitivo compreender e utilizar a automação da modelagem de ameaças, sem me sentir confuso?
	7	Facilidade	Eu considero fácil executar as tarefas necessárias para automação da modelagem de ameaças?
Intenção de Uso	8	Intenção de utilizar o sistema	Eu pretendo utilizar a automação da modelagem de ameaças no futuro para realizar minhas tarefas?
	9	Melhoria	Eu acredito que é melhor utilizar automação da modelagem de ameaças do que outros métodos manuais?
	10	Gostar do Sistema	Eu gosto de utilizar a automação da modelagem de ameaças para a realização das modelagens de ameaças?
Ajuste Tecnologia-Tarefa	11	Detalhe das Informações	A automação da modelagem de ameaças proposta oferece informações detalhadas para obter meus resultados na modelagem de ameaças?
	12	Localização das Informações	Quando necessito, na automação da modelagem de ameaças, eu localizo a informação de forma fácil e rápida?
	13	Atualidade das Informações	As informações fornecidas pela automação da modelagem de ameaças são atuais?
	14	Compreensão	As informações de que eu necessito são apresentadas pela automação da modelagem de ameaças de forma que facilita minha compreensão das ameaças?
	15	Confiabilidade	O resultado da automação da modelagem de ameaças é confiável?

Fonte: O Autor.

avaliativos e o fluxo operacional da abordagem, resultando em ajustes para maior precisão dos dados coletados.

3. Apresentação introdutória aos participantes sobre os conceitos fundamentais de modelagem de ameaças, visando assegurar entendimento homogêneo do domínio avaliado.

4. Orientações específicas sobre o exercício prático, esclarecendo as atividades a serem realizadas com a ferramenta durante a modelagem de ameaças. A descrição do sistema de exemplo está disponível no Apêndice E.

5. Elaboração do Diagrama de Fluxo de Dados (DFD) pelos participantes utilizando a ferramenta OWASP Threat Dragon, incluindo os principais elementos arquiteturais do sistema-alvo.

6. Rotulação dos elementos e fluxos de dados no DFD, padronizando termos para posterior comparação semântica no mecanismo de recomendação.

7. Conversão do modelo criado no Threat Dragon para o formato UTML, por meio do *Threat Copilot*, viabilizando a interpretação automatizada do modelo. Ingestão dos artefatos do modelo na base do *Threat Copilot*, permitindo sua análise pelo mecanismo de recomendação.

8. Execução do algoritmo de recomendação de ameaças, com apresentação das sugestões com base no grau de similaridade calculado para cada fluxo do DFD. Seleção e inspeção das ameaças recomendadas, permitindo ao usuário validar ou ajustar as proposições apresentadas.

9. Geração, visualização e edição do relatório final de ameaças, documentando os resultados da modelagem e possibilitando que o participante finalize sua análise.

10. Preenchimento do questionário *online* pelos participantes usando o Google Forms com as questões demográficas, variáveis do TAM-TTF e questões abertas.

Todas as interações com os participantes foram realizadas e gravadas por meio da ferramenta de comunicação *online* da Organização X Microsoft Teams (Microsoft Corporation, 2017).

4.2.1 Piloto para Validação do Instrumento de Coleta

O piloto teve como objetivo verificar a clareza, pertinência e adequabilidade das questões do questionário baseado nos construtos do modelo TAM e TTF. Buscou-se identificar dúvidas de interpretação, redundâncias e necessidade de ajustes antes da aplicação final do instrumento junto aos participantes do estudo principal.

O piloto contou com três participantes (denominados Piloto 1, Piloto 2 e Piloto 3), todos atuantes em áreas correlatas à segurança da informação. Os perfis foram selecionados intencionalmente para representar diferentes níveis de experiência com modelagem de ameaças e segurança da informação, o que permitiu observar como usuários com formações distintas interpretavam o questionário. Sobre a clareza e redundância das questões, o Piloto 1 considerou as perguntas “Eu dificilmente me confundo ao utilizar a automação da modelagem de ameaças?” (6) e “Eu considero a automação da modelagem de ameaças fácil de usar?” (7) como repetitivas, sugerindo que ambas capturavam o mesmo aspecto da facilidade de uso.

O Piloto 2 relatou dúvida quanto ao escopo da questão “As informações oferecidas pela automação da modelagem de ameaças proposta são de difícil localização?” (12), indicando incerteza sobre se a pergunta se referia à ferramenta em teste ou à disponibilidade de informações no mercado. Já o Piloto 3 destacou que o termo “confundir-se” poderia referir-se tanto às funcionalidades da ferramenta quanto aos conceitos teóricos da modelagem de ameaças, o que gerava ambiguidade.

No contexto da experiência dos participantes, o Piloto 1 apontou que, por não ter experiência prévia com automação, seria interessante incluir questões que permitissem expressar a intenção de uso futuro, e não apenas experiências passadas. O Piloto 2 considerou o questionário adequado. Já o Piloto 3 sugeriu eliminar redundâncias entre as questões sobre produtividade e velocidade. Os participantes recomendaram ampliar o escopo do questionário para contemplar: o conhecimento prévio do processo de modelagem, a clareza da relação entre análise de riscos e modelagem de ameaças; a identificação das funcionalidades mais relevantes da ferramenta, permitindo a priorização de aspectos percebidos como mais úteis. Também recomendaram padronizar os termos (“fácil” e “difícil”), numerar questões para evitar confusão e a manutenção do formato conciso do questionário.

Essas recomendações motivaram a inclusão de novas questões sobre o contexto organizacional, experiência anterior e avaliação de funcionalidades específicas. Com base nas observações, o questionário foi reformulado, incorporando novas variáveis contextuais:

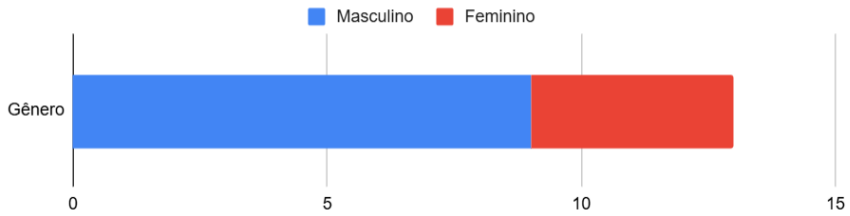
1. **Área de atuação:** “Qual área (divisão, departamento, etc.) da organização você trabalha?”.
2. **Conhecimento prévio sobre métodos e automação:** “Você conhecia (antes do contato atual) algum método de modelagem de ameaças? (STRIDE, LINDDUN, árvores de ataque, etc)” e “Você já utilizou algum tipo de automação para apoio na identificação de ameaças? (Qualquer tipo ferramenta, Threat Dragon, Microsoft TMT, STRIDEGPT, CHATGPT, GEMINI, etc.)”.
3. **Aspecto mais importante da ferramenta:** Questão fechada de múltipla escolha, incluindo integração com a ferramenta Threat Dragon (Conversão automática entre modelos), recomendação automática de ameaças (Automação), padronização via base de conhecimento de ameaças, categorização dos elementos com termos (tags) padronizadas, linguagem de descrição do diagrama e das ameaças (YAML) e gestão e repositório dos diversos modelos de ameaças.
4. **Melhoria de clareza nas perguntas:**
    - **Na variável simplicidade (6):** "Eu considero simples e intuitivo compreender e utilizar a automação da modelagem de ameaças, sem me sentir confuso?";
    - **Na variável Facilidade (7):** "Eu considero fácil executar as tarefas necessárias para automação da modelagem de ameaças?"
    - **Na variável Localização (12):** "Quando necessito, na automação da modelagem de ameaças, eu localizo a informação de forma fácil e rápida?".

O piloto possibilitou ajustes redacionais e melhorias na clareza semântica das perguntas e inclusão de novas variáveis que permitem uma interpretação sobre outros aspectos da experiência dos respondentes. Essas modificações apresentaram o questionário como instrumento em uma versão mais refinada para a fase de aplicação definitiva com os participantes.

4.2.2 Perfil dos Participantes da Avaliação

Os participantes foram selecionados por trabalharem nas áreas de segurança da informação da Organização X avaliada e todos possuem experiência prévia em atividades relacionadas à segurança da informação. O restante desta subseção descreve o perfil demográfico dos participantes.

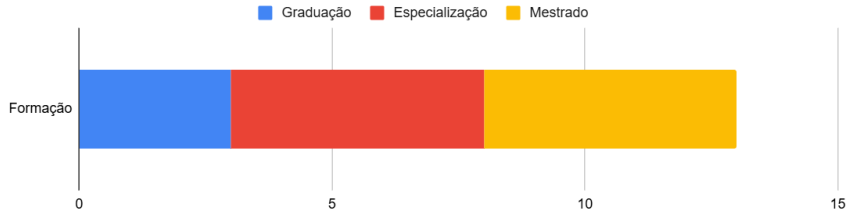
Figura 30 – Distribuição Participantes por Gênero



Fonte: O Autor

A Figura 30 apresenta a distribuição dos participantes da avaliação de acordo com o gênero. Observa-se uma predominância do gênero masculino, representando aproximadamente dois terços da amostra (nove participantes), enquanto o gênero feminino corresponde ao quantitativo restante (quatro participantes).

Figura 31 – Distribuição Participantes por Formação



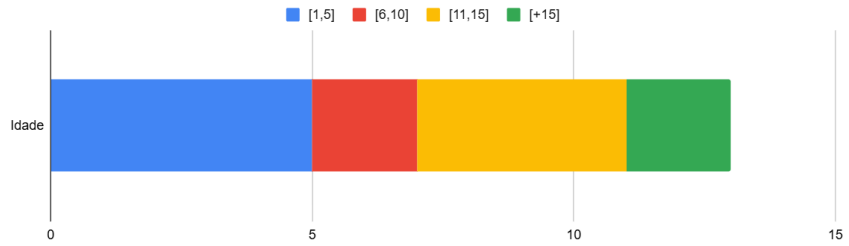
Fonte: O Autor

A Figura 31 apresenta a distribuição dos participantes de acordo com o nível de formação acadêmica. Observou-se a presença de três grupos distintos: graduação, especialização e mestrado. Ficando evidente a predominância de participantes com formação em nível de pós-graduação e mestrado, enquanto aqueles de graduação correspondem à menor proporção. Essa composição demonstrou um perfil de público com maturidade acadêmica, o que é relevante para o tipo de avaliação conduzida nesta pesquisa, pois participantes com maior nível de formação tendem a apresentar maior familiaridade com conceitos técnicos, metodologias de segurança e processos de engenharia de software.

A Figura 32 apresenta a distribuição dos participantes de acordo com a experiência em anos. Foram definidos quatro intervalos: [1–5], [6–10], [11–15] e [+15]. Percebeu-se uma concentração maior nas faixas intermediárias, especialmente entre 6 e 15 anos de experiência profissional, enquanto os grupos com até 5 anos e mais de 15 anos de experiência aparecem em menor número.

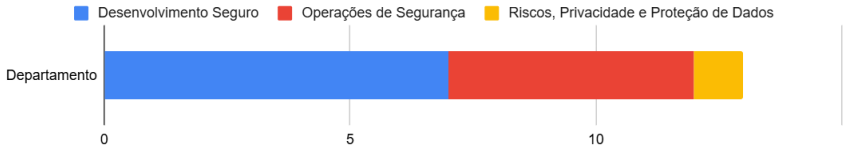
Essa distribuição evidencia um perfil de participantes composto por profissionais com experiência consolidada na área de segurança da informação, o que sugere a adequação do grupo à avaliação. O equilíbrio quantitativo quanto à presença de participantes menos experientes e mais experientes tende a contribuir para a análise comparativa da percepção de usabilidade e adequação tecnológica da ferramenta.

Figura 32 – Distribuição Participantes por Experiência



Fonte: O Autor

Figura 33 – Distribuição Participantes por Departamento de Segurança da Informação



Fonte: O Autor

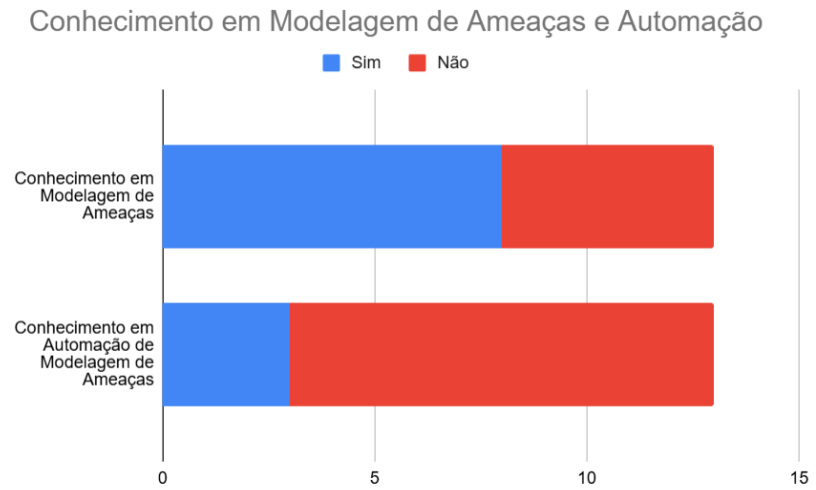
A Figura 33 apresenta a distribuição dos participantes conforme o departamento ou área de atuação profissional. Observa-se que a maioria pertence à área de Desenvolvimento Seguro, que é a responsável pelas práticas de segurança da informação para os times de desenvolvimento de software, inclusive de modelagem de ameaças. A segunda área foi a de Operações de Segurança, que é responsável pela infraestrutura de monitoramento de eventos e proteção contra ataques cibernéticos. Por fim, com um único participante teve a área de Riscos, Privacidade e Proteção de Dados, que é a área responsável pela gestão de riscos de segurança, gestão de incidentes de segurança e normativos de segurança.

A amostra contempla diferentes especializações dentro da área de segurança da informação da Organização X avaliada, possibilitando uma análise mais abrangente das percepções sobre a ferramenta de modelagem de ameaças automatizada. A presença significativa de sete profissionais de desenvolvimento seguro reforça a importância da ferramenta para apoiar práticas de segurança durante o desenvolvimento de software.

A Figura 34 apresenta a distribuição dos participantes em relação ao conhecimento prévio sobre modelagem de ameaças e automação de modelagem de ameaças. Observou-se que, pouco mais da metade dos participantes (8), afirmaram possuir algum nível de conhecimento em modelagem de ameaças, enquanto a outra parte declarou não ter experiência prévia. Por outro lado, quando o foco recai sobre o conhecimento em automação da modelagem de ameaças, ou seja, o uso de ferramentas e abordagens automatizadas para identificação e tratamento de ameaças, verificou-se uma predominância das respostas negativas. A maioria dos participantes não possui experiência anterior com soluções automatizadas.

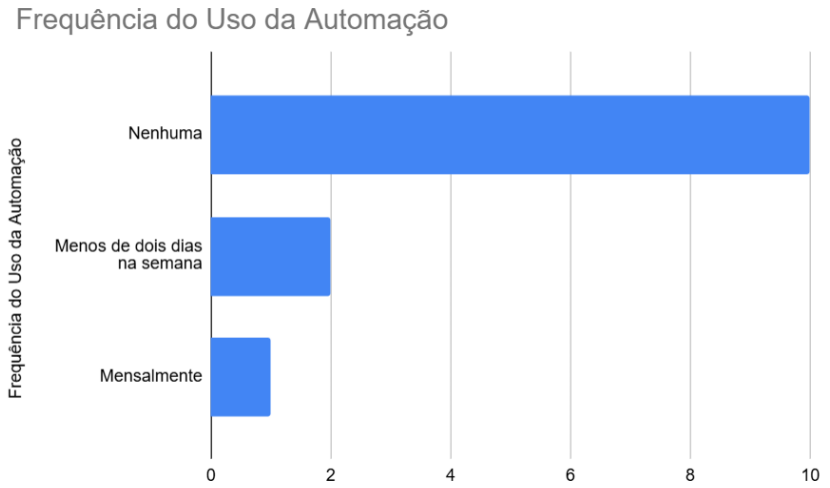
A Figura 35 apresenta a frequência de uso de ferramentas de automação entre os participantes do estudo. Observa-se uma predominância dos indivíduos (10) que não utilizaram nenhum tipo de automação em suas rotinas de trabalho, representando a maior parte da amostra. Em contraste, apenas uma pequena parcela relatou utilizar automação com alguma regularidade, sendo menos de dois dias por semana ou mensalmente.

Figura 34 – Distribuição dos Participantes Sobre o Conhecimento em Modelagem de Ameaças e Automação



Fonte: O Autor

Figura 35 – Frequência no Uso da Automação para Modelagem de Ameaças



Fonte: O Autor

Diante das respostas negativas sobre o conhecimento de soluções para automação da modelagem de ameaças, este resultado já havia sido previsto, e foi confirmado pelos dados da pesquisa.

4.2.3 Resultados Obtidos

Os dados coletados foram analisados de forma quantitativa visando avaliar a aceitação e adequação da modelagem automatizada de ameaças pela ferramenta. Para avaliar as questões, utilizou-se a abordagem definida por Cardoso e Cleophas (2020), na qual se calcula o Ranking Médio Individual (R.M.) dos itens da Escala Likert de cada uma das variáveis que constituíam os quatro construtos avaliados relacionados aos modelos TAM e TTF. O Ranking Médio considera a frequência das respostas de forma ponderada, considerando que quanto mais próximo de 5 (cinco) maior o nível de satisfação com a variável, enquanto valores próximos de 1 (um) indicam baixa concordância. A Equação 17 define o cálculo do R.M.:

$$R.M. = \frac{\sum f_i \times v_i}{NS} \tag{17}$$

A Equação 17 Ranking Médio é obtido a partir da soma ponderada das respostas, em que cada valor da escala (Vi) é multiplicado pela sua frequência observada (fi). Essa soma representa a Média Ponderada (MP), que agrega tanto a quantidade quanto a intensidade das respostas. Em seguida, essa soma ponderada é dividida pelo número total de respondentes (NS).

Na sequência, foi determinada a média aritmética entre os RM dessas variáveis, obtendo, assim, o RM de cada construto avaliado. Adicionalmente, foi calculado o desvio padrão das variáveis, para avaliar a variação das respostas. A Tabela 26 mostra a frequência das respostas dadas, o desvio padrão e os Rankings Médios calculados.

De forma geral a média de todos os construtos foi 4,30. A média de desvio padrão geral foi 0,68 indicando convergência na avaliação positiva feita pelos participantes.

O construto de maior valor foi a “Utilidade Percebida” com a média 4,73 e o desvio padrão de 0,45, onde os participantes consideram a automação da modelagem de ameaças útil para suas atividades. Já o construto de menor valor foi a “Facilidade de Uso” com a média 3,85 e o desvio padrão 0,84. A Figura 36 apresenta a média do RM de cada construto, calculada a partir do RM de suas respectivas variáveis.

Ao analisar individualmente o construto “Utilidade Percebida”, o maior RM (4,92) encontrado refere-se à variável “Utilidade”, em que os respondentes consideraram que a utilidade da ferramenta para suas tarefas. Já o menor RM (4,69) refere-se à variável “Desempenho”, indicando comparativamente uma dúvida sobre o impacto no desempenho no dia a dia dos profissionais participantes. De forma individual, foi realizada a análise de cada variável que compõe o construto “Utilidade Percebida”, permitindo o cálculo do RM correspondente a cada uma delas. Os resultados estão apresentados na Figura 37.

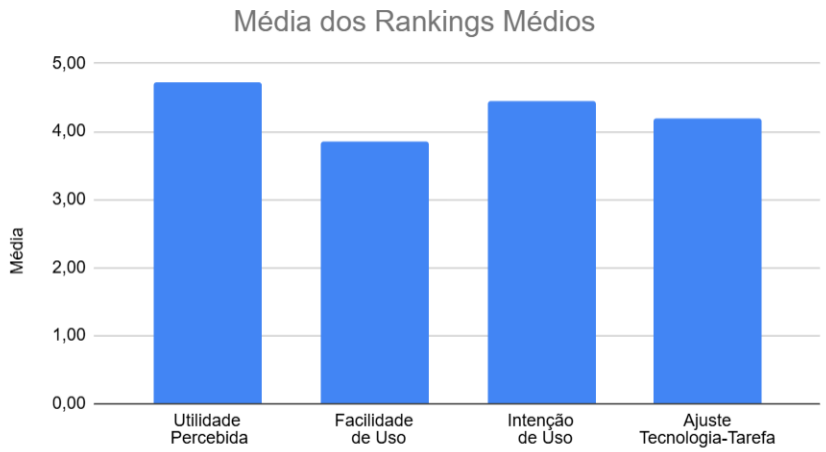
Em relação ao construto “Facilidade de Uso”, observou-se a menor média e o maior desvio padrão nas respostas, indicando comparativamente uma dificuldade no uso da ferramenta. Quanto às variáveis que constituem o construto, o maior RM de 4,08 e desvio padrão 0,73 refere-se à “Operação do método”. A variável “Facilidade” foi a com menor RM de 3,69 e desvio padrão 0,99, ela avaliou a

Tabela 26 – Resultados dos Questionários TAM–TTF

Construto	**ID**	**Variável**	**D.T.**	**D.P.**	**N.C. N.D.**	**C.P.**	**C.T.**	**R.M.**	**Des. P.P.**	**Média R.M.**	**Média Des. P. P.**
Utilidade Percebida	1	Desempenho	0	0	1	4	8	4,54	0,63	4,73	0,45
	2	Produtividade	0	0	0	3	10	4,77	0,42		
	3	Rapidez	0	0	0	4	9	4,69	0,46		
	4	Utilidade	0	0	0	1	12	4,92	0,27		
Facilidade de Uso	5	Operação do Método	0	0	3	6	4	4,08	0,73	3,85	0,84
	6	Simplicidade	0	1	4	5	3	3,77	0,89		
	7	Facilidade	0	2	3	5	3	3,69	0,99		
Intenção de Uso	8	Intenção de utilizar o sistema	0	0	2	4	7	4,38	0,74	4,44	0,70
	9	Melhoria	0	0	0	5	8	4,62	0,49		
	10	Gostar do Sistema	0	0	3	3	7	4,31	0,82		
Ajuste Tecnologia Tarefa	11	Detalhe das Informações	0	0	2	4	7	4,38	0,74	4,18	0,76
	12	Localização das Informações	0	0	8	2	3	3,62	0,84		
	13	Atualidade das Informações	0	0	3	6	4	4,08	0,73		
	14	Compreensão	0	0	2	3	8	4,46	0,75		
	15	Confiabilidade	0	0	2	4	7	4,38	0,74		

Fonte: O Autor.

Figura 36 – Médias do Construtos do TAM TTF

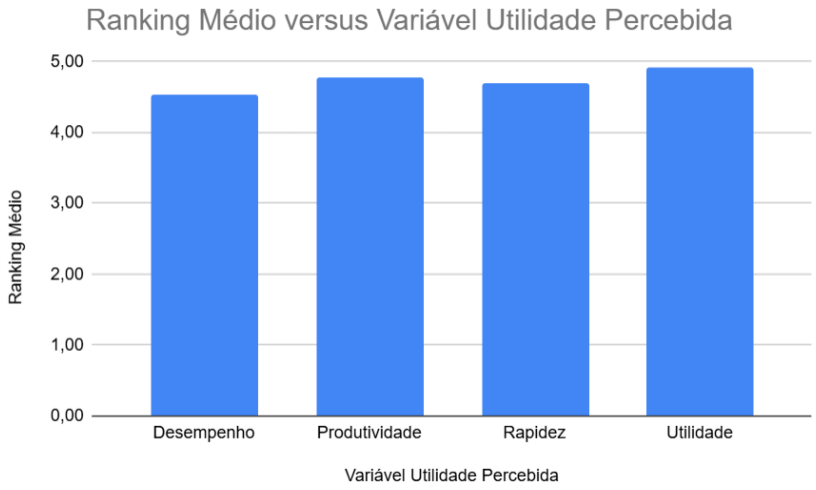


Fonte: O Autor

facilidade na execução das tarefas para a modelagem de ameaças automatizada. A Figura 38 apresenta os Rankings Médios do construto “Facilidade de Uso”.

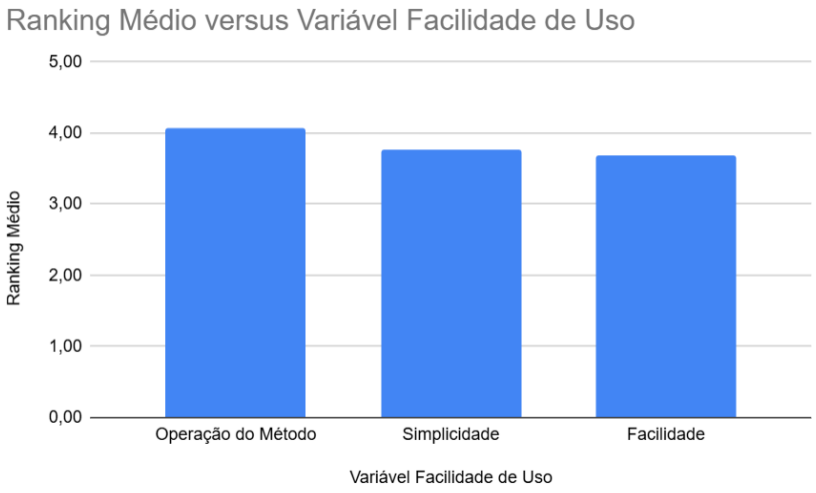
Para o construto “Intenção de Uso”, obteve-se um RM de 4,44 e desvio padrão de 0,70. A variável com maior pontuação foi “Melhoria”, com RM, 4,62 e desvio padrão de 0,49, onde observou-se se a automação da ferramenta melhora a modelagem de ameaças. Por outro lado, a variável com menor pontuação foi “Gostar do Sistema”, com RM de 4,31 e desvio padrão de 0,82, que se refere à percepção

Figura 37 – Resultados da Variável Utilidade Percebida



Fonte: O Autor

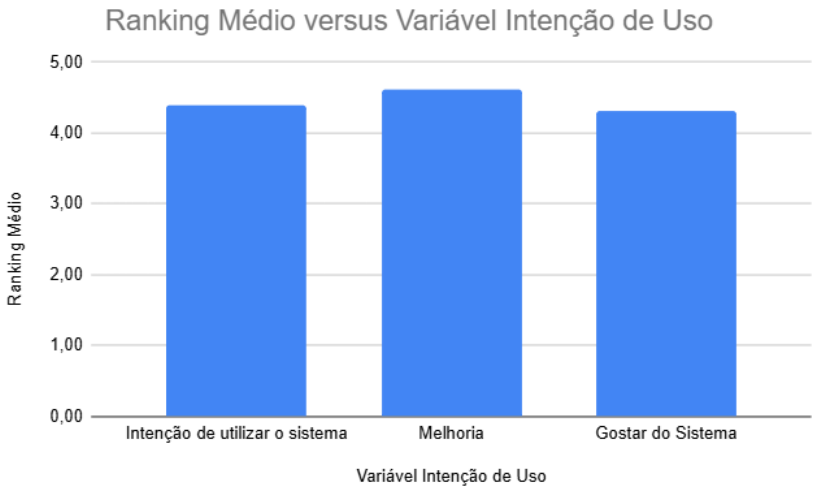
Figura 38 – Resultados da Variável Facilidade de Uso



Fonte: O Autor

de gostar do sistema automatizado para a modelagem de ameaças. A Figura 39 apresenta os resultados obtidos das variáveis para o construto “Intenção de Uso”.

Figura 39 – Resultados da Variável Intenção de Uso

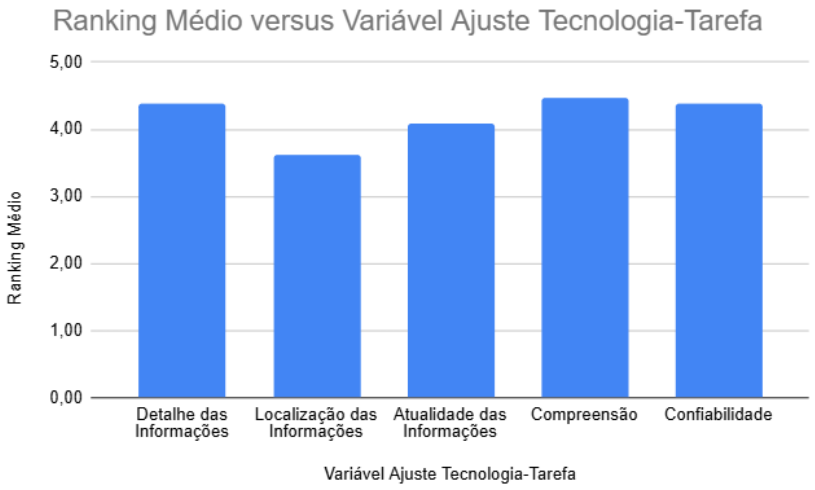


Fonte: O Autor

Por fim, em relação ao construto “Ajuste Tecnologia-Tarefa” que visa avaliar o ajuste da automação da modelagem de ameaças às necessidades da modelagem de ameaças, obteve-se um RM de 4,18 e desvio padrão de 0,76, que se aproxima da resposta “concordo parcialmente”.

A variável com maior RM de 4,46 e desvio padrão de 0,75 foi a “Compreensão”, onde observou-se a facilidade no entendimento das ameaças por parte do usuário do sistema. A variável com menor desempenho foi a de “Localização das Informações”, inclusive entre todas as variáveis dos demais construtos, com RM de 3,62 e desvio padrão de 0,8. Esta variável investiga se as informações obtidas são localizadas de forma fácil e rápida. A Figura 40 mostra os resultados obtidos para a variável TTF.

Figura 40 – Resultados da Variável TTF



Fonte: O Autor

Para as questões qualitativas abertas utilizou-se o agrupamento por meio de conceitos envolvidos nas respostas, as respostas originais podem ser vistas no Apêndice F. Na primeira questão aberta sobre o que cada entrevistado achou da proposta da automação da modelagem de ameaças, as respostas se concentraram em três conceitos relevantes:

1. **Utilidade e valor agregado:** Diversas respostas sugerem ser uma proposta válida, promissora e que deve ser utilizada pelos analistas de segurança. Termos como "muito interessante", "promissor" e "sensacional" apareceram de forma recorrente, sugerindo entusiasmo com o conceito da solução. A possibilidade de gerar relatórios detalhados, apresentar recomendações de ameaças de forma automatizada e controles associados, e permitir a seleção ou exclusão de ameaças indicadas foram aspectos apontados como importantes.

2. **Base de Conhecimento e Intuitividade:** Algumas respostas mencionaram a importância da base de conhecimento de ameaças, a importância da integração com a ferramenta gráfica de Diagrama de Fluxo de Dados e a intuitividade e praticidade no uso da ferramenta.

3. **Aprimoramento e Especialistas:** Alguns participantes ressaltaram a necessidade de testes complementares e da importância de um time de especialistas para tirar melhor proveito das capacidades da ferramenta. Também foi apontado o custo em tempo da atividade de modelagem e rotulação dos elementos do diagrama de fluxo de dados e a importância da padronização das ameaças e controles de segurança via base de conhecimento.

A segunda questão aberta buscou identificar as oportunidades de melhoria na proposta apresentada para os entrevistados. As propostas observadas foram agrupadas em quatro conceitos relevantes:

1. **Melhor suporte ao uso dos termos:** Os entrevistados ressaltaram a importância de simplificar, até mesmo através da seleção automática de termos por contexto, a fase de rotulação dos elementos do diagrama de fluxo de dados.

2. **Aprimoramento da usabilidade:** Entre as sugestões mais recorrentes, destaca-se a possibilidade de navegar dentro de uma "página do modelo", permitindo acessar elementos, relatórios e recomendações sem precisar retornar à tela principal. Também foi mencionada a integração com outras ferramentas de modelagem e a melhoria da interface do Threat Dragon diante das dificuldades encontradas no seu uso. Também, foi sugerida a disponibilização de uma versão da ferramenta avaliada em português do Brasil. Tais aspectos sugeriram a importância da experiência do usuário e da interoperabilidade no ambiente de trabalho do analista de segurança da informação.

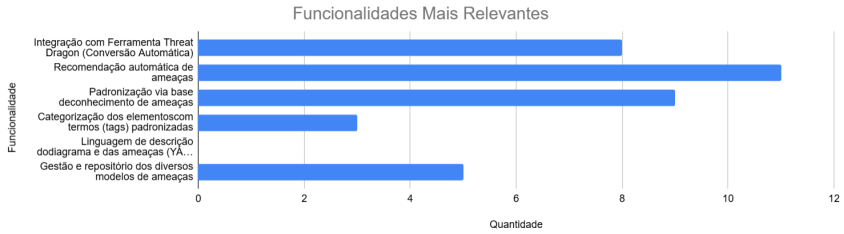
3. **Aprimoramento das ameaças e relatório:** Diversos comentários destacaram a importância de gerar estatísticas sobre as ameaças, como frequência, recorrência por tipo de aplicação ou severidade. Além disso, foi sugerida a inclusão do diagrama de fluxo de dados no relatório final, bem como a padronização textual das descrições das ameaças e a inserção de categorias na listagem, facilitando a busca e a interpretação. Essas melhorias visam aumentar a clareza e a utilidade das informações apresentadas nos resultados da modelagem.

4. **Considerações temporais sobre os modelos de ameaças:** Sugeriu-se que os modelos de ameaças antigas tenham peso reduzido nas análises automáticas, evitando que informações desatualizadas

influenciem recomendações futuras. Também foi mencionada a necessidade de realimentar o sistema com vulnerabilidades já tratadas, de modo que estas não voltem a aparecer em relatórios subsequentes. Tal funcionalidade contribuiria para manter o repositório coerente e atualizado com o estado real da segurança do sistema.

Por fim, foi questionado aos participantes quais as funcionalidades consideradas mais relevantes na ferramenta de modelagem de ameaças automatizada. Observa-se que as três funcionalidades com maior destaque foram: recomendação automática de ameaças, padronização via base de conhecimento de ameaças e integração com a ferramenta *OWASP Threat Dragon* (conversão automática), todas recebendo um número expressivo com onze, nove e oito menções, respectivamente. A Figura 41 apresenta as respostas dos entrevistados. Observou-se também que nenhum dos participantes mencionou a linguagem de descrição de ameaças UTML como relevante.

Figura 41 – Funcionalidades Mais Relevantes para os Participantes



Fonte: O Autor

4.3 Avaliação Empírica *Online* de Utilização Real da Ferramenta

A modelagem de ameaças é uma prática essencial para prevenir vulnerabilidades de segurança desde a concepção inicial do sistema. Até então, na Organização X avaliada, o procedimento de modelagem de ameaças era feito de forma manual e não estruturado, sendo responsabilidade do analista de segurança definir qual método utilizar, o que representava um grande esforço na identificação, revisão e seleção de controles de segurança.

Este relato descreve a utilização prática e real da ferramenta de modelagem de ameaças automatizada descrita nesta pesquisa, destacando as observações mais relevantes para o contexto da elicitação automática das ameaças. De acordo com (GARCÍA, 2024), avaliações *online* em sistemas de recomendação, quando usuários reais realizam atividades no sistema, geram feedbacks relevantes para a melhoria do sistema.

O cenário de aplicação foi a modelagem de ameaças para um novo produto de software que objetiva contribuir no contexto da educação brasileira e ainda para a oferta de benefícios sociais em atendimento uma nova lei aprovada em 2025. A demanda de trabalho foi solicitada pelo Encarregado pela Privacidade e Proteção de Dados Pessoais (DPO - *Data Protection Officer*) da Organização X para avaliar a segurança do software que está sendo desenvolvido. Este produto tem um alcance potencial inicial de 3 milhões de usuários e terá dados pessoais, biométricos e profissionais de um grupo de indivíduos da

população brasileira. A modelagem de ameaças foi realizada em outubro de 2025. As características dos participantes estão listadas na Tabela 27:

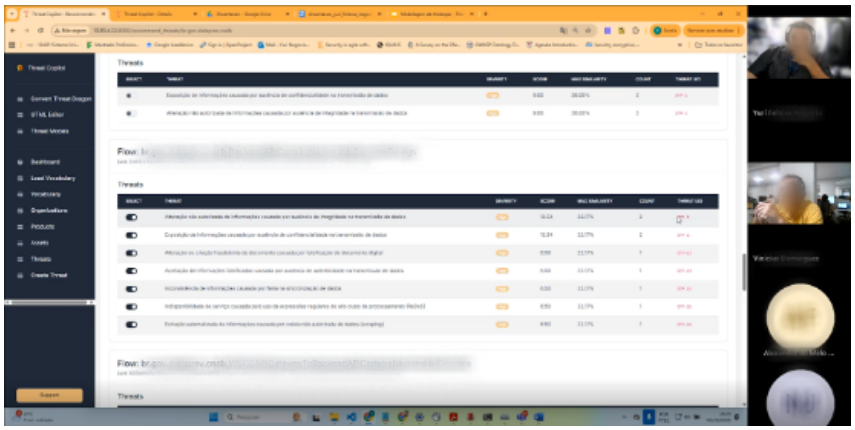
Tabela 27 – Perfis Profissionais do Uso Real da Ferramenta

Participante	**Perfil profissional**	**Experiência**
Participante 1	Analista de Segurança	15 anos
Participante 2	Analista de Segurança	12 anos
Participante 3	Analista de Segurança	11 anos
Participante 4	Analista de Segurança	11 anos
Participante 5	Analista de Segurança	8 anos
Participante 6	Arquiteto de Software	18 anos
Participante 7	Arquiteto de Software	6 anos
Participante 8	Arquiteto de Software	22 anos
Participante 9	Desenvolvedor	22 anos

Fonte: O Autor.

Esta atividade foi realizada a partir da participação e observação direta do pesquisador no uso da ferramenta durante quatro sessões através da ferramenta de comunicação e colaboração Microsoft Teams (Microsoft Corporation, 2017). A Figura 42 apresenta a tela de um das reuniões com os analistas de segurança para avaliação da *Threat Copilot*.

Figura 42 – Reunião de Modelagem de Ameaças usando a *Threat Copilot* realizada através do Microsoft Teams



Fonte: O Autor

A primeira sessão foi realizada em uma reunião *online* (2h:32min) com os participantes presentes nos seguintes estados: Ceará, Paraíba, Santa Catarina e Rio Grande do Norte. A segunda sessão foi uma revisão do modelo de ameaças com apenas dois participantes (43 min), esta sessão foi motivada pela impossibilidade de um dos participantes estar disponível no momento da primeira sessão. A terceira sessão foi realizada com (1h:25 min) todos os analistas de segurança. Por fim, a última reunião foi realizada com o "dono do produto", que é o responsável pela gestão e desenvolvimento do produto, para apresentar o relatório (60 min). A modelagem de ameaças ocorreu conforme as seguintes etapas:

1. Kickoff da modelagem de ameaças (4 minutos);

2. Sessão de modelagem do DFD utilizando a ferramenta OWASP Threat Dragon com participação dos arquitetos e desenvolvedores do produto (1h:36min);
3. Conversão do diagrama DFD gerado pela OWASP Threat Dragon para a *Threat Copilot* (5min);
4. Execução das recomendações automáticas de ameaças (10s);
5. Análise e revisão dos resultados das recomendações (48min);
6. Inclusão de novas ameaças (4min);
7. Revisão do diagrama e das ameaças selecionadas (43min);
8. Revisão dos controles de segurança (1h:25min);
9. Emissão do relatório e reunião de entrega para o dono do produto (1h).

O *kickoff* da modelagem de ameaças levou 4 minutos, em que foram apresentadas a origem da solicitação e as características do novo sistema. Em seguida, a sessão de modelagem do diagrama de fluxo de dados durou 1h:36min e se iniciou com a leitura do artefato do documento de arquitetura de software. Este documento definiu a proposta do sistema em termos de componentes (servidores, api gateways, bancos de dados, etc), de funcionalidades de negócio e de atores do sistema utilizando uma descrição textual e gráfica baseada no padrão da indústria Archimate (The Open Group, 2022). Algumas dúvidas dos analistas de segurança surgiram durante a sessão de modelagem do diagrama de fluxo de dados e os arquitetos e um dos desenvolvedores foram convidados para explicar a proposta com maior riqueza de detalhes. Após a explicação, o diagrama de fluxo de dados foi finalizado e seus elementos e fluxos foram rotulados com os termos definidos no vocabulário da ferramenta.

Após o fim da definição do DFD, foi realizada a fase de elicitação das ameaças. Neste momento, a ferramenta recomendou as ameaças mais similares para cada fluxo de dados do modelo de ameaças. Para cada ameaça definida no fluxo, os analistas de segurança avaliaram a pertinência ou não para o modelo definido. Durante a fase de elicitação de ameaças, percebeu-se a necessidade de remover quatro fluxos de dados vistos como desnecessários por estarem fora do escopo de trabalho do produto. Ao final desta avaliação, foram observados os seguintes dados apresentados na Tabela 28:

A Tabela 28 apresenta um resumo dos resultados obtidos no uso real da ferramenta *Threat Copilot*. Ela descreve as características do diagrama avaliado (como a quantidade de fluxos, elementos e termos do vocabulário), os números associados ao processo de recomendação de ameaças, incluindo ameaças recomendadas, válidas, não selecionadas e novas ameaças identificadas, e os valores finais de *precision*, *recall* e *f1-measure* calculados a partir desses dados. Por fim, registra também o total de controles de segurança sugeridos pela ferramenta.

No cálculo das métricas *precision* e *recall* as três novas ameaças não foram consideradas como falsos negativos, pois não pertenciam ao conjunto de dados e não tinham a possibilidade de serem recomendadas. Em relação a necessidade de novas ameaças, foi observado que elas estão relacionadas às novas exigências para atendimento da Lei Geral de Proteção de Dados Pessoais (Brasil, 2018). As três novas ameaças utilizadas estão descritas na Tabela 29.

Tabela 28 – Dados do uso real da ferramenta

Observação	**Unidade de Análise**	**Dados**
Total de Fluxos de Dados	DFD	11
Total de Elementos (Atores, Processos, Armazenamentos)	DFD	11 (2 atores, 4 processos e 5 armazenamentos)
Total de termos distintos utilizados	Vocabulário	11
Total de ameaças recomendadas pela ferramenta	Recomendação de Ameaças	77
Total de ameaças não selecionadas (Falsos Positivos – FP)	Recomendação de Ameaças	31
Total de ameaças recomendadas válidas (Positivo Verdadeiro – PV)	Recomendação de Ameaças	46
Total de ameaças incluídas (Falsos Negativos – FN)	Recomendação de Ameaças	15
Total de novas ameaças não existentes na base de conhecimento	Recomendação de Ameaças	3
Total de vezes da escolha das novas ameaças	Recomendação de Ameaças	4
Total de ameaças finais	Recomendação de Ameaças	64
Total de ameaças distintas	Recomendação de Ameaças	18
Precisão (PV/(FP+PV))	Avaliação	60%
recall (PV/(PV+FN))	Avaliação	75%
F1-Score ($2 \times \frac{P \times R}{P+R}$)	Avaliação	67%
Total de controles de segurança recomendados	Controles de Segurança	50

Fonte: O Autor.

Tabela 29 – Novas Ameaças Identificadas

ID	**Descrição da nova ameaça**
DTP-35	Exposição ou vazamento de dados pessoais sensíveis devido a acesso lógico indevido ao banco de dados.
DTP-36	Exposição indevida de dados pessoais causada pelo registro de informações sensíveis em logs de aplicação, auditoria ou monitoramento.
DTP-37	Exposição indevida de dados sensíveis armazenados em disco, *backups* ou mídias removíveis causada por roubo, cópia não autorizada ou comprometimento físico do banco de dados.

Fonte: O Autor.

Ressalta-se que a LGPD embora tenha sido sancionada em 2018, só se tornou obrigatória em setembro de 2020. Desta forma, os modelos de ameaças da base de conhecimento não consideraram as ameaças específicas com impacto na privacidade dos dados pessoais. Espera-se que, à medida que os novos modelos de ameaças forem sendo elaborados, a ferramenta recomende também tais ameaças, melhorando as métricas de avaliação.

Ao final do processo, o relatório de ameaças e controles de segurança foi distribuído para as áreas de privacidade, gestão de vulnerabilidades, *security operation center* (SOC) e para o dono do produto de

software para que os planos de tratamento sejam executados.

5 DISCUSSÃO DAS AVALIAÇÕES

Este capítulo é dedicado à análise e discussão dos resultados obtidos, explorando em profundidade a resposta à questão de pesquisa. Também são apresentadas as discussões sobre os resultados obtidos nas avaliações *online* e *offline* realizadas.

5.1 Discussão da Questão de Pesquisa

A questão de pesquisa definida na seção 1.2 para este estudo foi: *"Como automatizar o processo de modelagem de ameaças em softwares utilizando o conhecimento prévio de modelos de ameaças?"*. Realizou-se inicialmente uma busca na Organização X avaliada com o objetivo de identificar os modelos de ameaças previamente elaborados pelas equipes de segurança da informação. A partir desses modelos, apresentados em dados não estruturados, analisaram-se suas características e a relação com o método de modelagem de ameaças baseado no STRIDE. Após essa avaliação, foram desenvolvidos modelos estruturados para representar os modelos de ameaças (UTML), o vocabulário utilizado na forma de uma WordNet específica do domínio e a base de conhecimento que descreve as ameaças e seus respectivos controles.

A partir disso, foi elaborado um modelo de comparação entre fluxos de dados de um diagrama de fluxo de dados, denominado Unidade de Ameaça. Esse modelo permitiu avaliar o grau de similaridade entre um fluxo de dados analisado e fluxos de dados previamente registrados na base histórica de ameaças. Ao todo, foram identificadas 103 unidades de ameaça nos 34 modelos históricos de ameaças selecionados.

Após a definição da unidade de similaridade foi elaborado o mecanismo de recomendação para comparar baseado na similaridade e em pesos, definidos a partir do consenso de especialistas, para determinar quais as unidades de ameaças da base de conhecimento são as mais adequadas para uma unidade de ameaça consultada.

Para comprovar o seu funcionamento foi desenvolvida a ferramenta *Threat Copilot* que possui um sistema de recomendação baseado em conteúdo no qual compara unidades de ameaças de um novo modelo com as unidades de ameaças da base de conhecimento. A ferramenta recomenda as ameaças a partir de um diagrama de fluxo de dados visual gerado por uma ferramenta de modelagem de ameaças (*OWASP Threat Dragon*), sugerindo quais ameaças cada fluxo de dados do diagrama possui, permitindo que o analista de segurança selecione as ameaças válidas, adicione novas ameaças e gerencie seus modelos.

Ao final, os modelos de ameaças foram inseridos na ferramenta, possibilitando a realização das avaliações *offline* e *online* do processo de modelagem de ameaças. Os resultados obtidos sugerem que é viável empregar técnicas de sistemas de recomendação para apoiar a modelagem de ameaças de forma automatizada, a partir da comparação entre unidades de ameaça.

Sendo assim, comprovando que é possível, esta pesquisa propôs e implementou uma abordagem prática baseada em um sistema de recomendação que, por meio da navegação em um grafo derivado do diagrama de fluxo de dados (DFD) de um novo modelo de ameaças, identifica e sugere ameaças

potencialmente relevantes para novas arquiteturas a partir do reuso de conhecimento contido em modelos de ameaça históricos. As recomendações são obtidas pela comparação de similaridade entre o novo modelo e o repositório de modelos anteriores, considerando etiquetas, tipos de elementos e relacionamentos. Para a análise de etiquetas, adotou-se um mecanismo de similaridade semântica baseado no algoritmo de Wu e Palmer (1994), apoiado por uma WordNet especializada no domínio de modelagem de ameaças.

As próximas seções discutem os resultados obtidos nas avaliações realizadas na ferramenta *Threat Copilot*.

5.2 Discussão da Avaliação *Offline*

Quando consideramos a dimensão da corretude na avaliação de um sistema de recomendação de ameaças buscamos avaliar o quanto as recomendações das ameaças são corretas e abrangentes. A principal vantagem dos experimentos *offline* é o baixo custo, pois podem ser executados sempre que necessário. Além disso, em comparação com outras abordagens, experimentos *offline* são mais rápidos e podem ser conduzidos em um ambiente controlado (GARCÍA, 2024). No contexto da avaliação *offline*, os resultados obtidos sobre as métricas de *precision*, *recall* e *f1-measure* foram 51%, 72% e 56%. Indicando que é necessário evoluir o sistema de recomendação da ferramenta para que ele seja mais preciso e que garanta uma maior cobertura das ameaças recomendadas.

A métrica de *precision*, que mede o percentual de acertos, indicou que, em média, apenas cerca de metade das ameaças recomendadas pela ferramenta são, de fato, ameaças que o analista de modelagem de ameaças consideraria válidas e relevantes. A outra metade são falsos positivos (FP), ou seja, ameaças irrelevantes que a ferramenta sugeriu, o que pode levar a um desperdício de tempo do analista de segurança da informação revisando recomendações inadequadas.

A métrica de *recall*, que mede a proporção de ameaças relevantes que a ferramenta recomendou entre todas as ameaças realmente relevantes no conjunto de dados, indicou que a ferramenta consegue identificar e recomendar 72% das ameaças importantes. Os falsos negativos (FN), ameaças relevantes que o sistema não recomendou, representam cerca de 28% das ameaças. No contexto de segurança, o percentual de FN significa o percentual de ameaças reais e importantes estão sendo perdidas pelo processo de modelagem de ameaças, resultando em riscos reais ao sistema avaliado por sua ausência de mitigação.

A métrica de *f1-measure*, que fornece uma visão sobre o *trade-off* entre a *precision* e *recall*, apresentou um valor relativamente baixo e reflete o desempenho moderado do sistema, indicando que há um espaço significativo para melhoria tanto na precisão das recomendações (reduzindo irrelevâncias) quanto na completude (reduzindo ameaças perdidas).

Quando foi observada a validação cruzada das métricas (K-Fold) ao longo das cinco execuções, percebeu-se que o *recall* apresentava valores superiores aos de *precision* em praticamente todos os folds. Esse comportamento sugere que o sistema de recomendação conseguiu recuperar uma parte das ameaças relevantes existentes no conjunto de referência (72%). Observou-se também que existe variação significativa dos resultados obtidos em cada *fold*, o que sugere que o desempenho da ferramenta pode ser sensível no resultado das métricas ao subconjunto de unidades de ameaça usado para treinamento e testes. Esta sensibilidade é esperada, principalmente, por conta da quantidade reduzida de modelos de ameaças

utilizados na base de conhecimento e a variabilidade destes modelos.

Analisando o contexto da avaliação *offline*, a ferramenta atual possui dificuldades com falsos positivos, exigindo um maior custo do analista de segurança da informação para avaliar e descartar ameaças irrelevantes, e moderadamente com falsos negativos, exigindo que o mesmo analista considere ameaças não recomendadas.

É importante observar, ainda, que o tamanho reduzido do *dataset* inicial constitui uma limitação relevante tanto para a generalização quanto para o desempenho das métricas das recomendações produzidas pela *Threat Copilot*.

A base de conhecimento inicial foi composta por apenas 34 modelos de ameaças e 103 unidades de ameaça, o que resultou em uma cobertura limitada dos diferentes diagramas de fluxo de dados utilizados na modelagem de ameaças. A isso soma-se a baixa diversidade na rotulação dos elementos e fluxos de dados, evidenciada, por exemplo, pela predominância do termo *"http"* para representar fluxos de dados e do termo *"application server"* para representar processos, conforme ilustrado na Figura 17. Adicionalmente, verificou-se que 59% dos modelos possuíam até 15 ameaças, indicando que uma parcela significativa dos registros históricos apresenta baixa densidade de ameaças por unidade analisada.

Como consequência, o recomendador tende a operar sobre um conjunto reduzido de unidades de ameaça comparáveis, muitas delas bastante semelhantes entre si, o que impacta diretamente as métricas de *precision* e *recall*. Em relação ao *recall*, uma base histórica pequena e com cobertura limitada reduz a capacidade do sistema de recuperar todas as ameaças relevantes, aumentando a incidência de falsos negativos. Quanto à *precision*, a escassez e a baixa diversidade das unidades de ameaça tornam o processo de recomendação mais suscetível a associações imprecisas, elevando a ocorrência de falsos positivos. No contexto avaliado, esse cenário contribui para explicar os resultados da avaliação offline, em que se observou *recall* superior à *precision*, indicando que o sistema consegue recuperar ameaças relevantes, mas ainda apresenta limitações quanto à precisão das recomendações geradas.

Esse conjunto de fatores ajuda a compreender os resultados mais modestos obtidos na avaliação das métricas. Assim, espera-se que a ampliação da base de conhecimento, com um número maior de modelos, maior diversidade no uso dos termos e maior densidade de ameaças por modelo, contribua para a melhoria das métricas e para a geração de recomendações mais equilibradas e confiáveis.

5.3 Discussão da Avaliação *Online* Centrada no Usuário

A integração dos modelos TAM e TTF permite compreender em que medida as características da tecnologia e o seu ajuste à tarefa são determinantes para o usuário optar por uma determinada ferramenta (DISHAW; STRONG, 1999) utilizando os construtos utilidade percebida, facilidade de uso percebida, intenção de uso e ajuste tecnologia-tarefa.

No contexto desta pesquisa, a utilização de uma ferramenta de automação de modelagem de ameaças está relacionada à percepção de sua utilidade e de sua facilidade de uso. A utilidade percebida é definida como o grau em que uma pessoa acredita que, ao utilizar determinado sistema, poderá melhorar seu desempenho. Já a facilidade de uso percebida refere-se ao grau em que a pessoa acredita que o uso do sistema reduz o esforço físico ou mental necessário para a execução de suas atividades. O ajuste entre

tecnologia e tarefa indica que a tecnologia tende a ser utilizada na medida em que atende às exigências de uma determinada tarefa, influenciando diretamente a utilidade percebida, a intenção de uso e a própria facilidade de uso (CARDOSO; CLEOPHAS, 2020).

Ao analisar os resultados, a utilidade percebida destacou-se como o construto mais bem avaliado. A variável 'Utilidade' atingiu média de 4,92, evidenciando que os participantes concordaram quase integralmente com a utilidade da proposta. As demais variáveis de desempenho, produtividade e rapidez também tiveram um bom desempenho, com os valores 4,54, 4,77 e 4,69, respectivamente. Nas respostas às questões abertas, os participantes enfatizaram de maneira recorrente a utilidade e o valor agregado da proposta, destacando, em particular, a relevância do mecanismo de recomendação e da base de conhecimento na padronização dos modelos de ameaça. Este resultado indica a percepção dos participantes quanto à perspectiva de melhoria da produtividade das suas atividades de modelagem de ameaças com o uso da ferramenta proposta. O referido resultado está diretamente conectado à importância de compartilhar os artefatos dos modelos de ameaça de forma padronizada para o reuso, educação e *benchmark*, e da crescente busca por automação na modelagem de ameaças (BERNSMED et al., 2022).

A facilidade de execução da modelagem de ameaças é um elemento relevante que serve como evidência para uma potencial adoção da prática da modelagem de ameaças (TRAN; YSKOUT; JOOSEN, 2023). Em relação à "Facilidade de Uso", observou-se uma dificuldade dos participantes em utilizar a ferramenta. O construto recebeu a menor nota (3,85) entre os demais construtos e o maior dissenso entre os participantes, com o valor 3,85 e a maioria das respostas (67%) entre "Não Concordo e Nem Discordo" e "Concordo Parcialmente", além do desvio padrão de 0,84. Observando o perfil dos participantes, tem-se que apenas um pouco mais da metade (61,5%) deles conheciam a modelagem de ameaças e poucos (23,1%) usaram algum tipo de automação para modelagem de ameaças. Essa constatação reitera a importância e necessidade de capacitações mais aprofundadas dos analistas de segurança em métodos de modelagem de ameaças, na própria ferramenta *Threat Copilot* e nas ferramentas auxiliares de diagramação, como a OWASP Threat Dragon. Os participantes também ressaltaram a necessidade de aprimoramento da ferramenta *Threat Copilot*, como, por exemplo, no uso dos termos, de evitar recomendações antigas e na importância do seu uso por especialistas para tirarem maior proveito e melhorarem a sua produtividade. Quanto às potenciais melhorias, eles indicaram a necessidade da melhoria da usabilidade da ferramenta *Threat Copilot*, como, por exemplo, facilitar a consulta aos termos do vocabulário e permitir a busca das ameaças de por categorias. Já para a ferramenta auxiliar OWASP Threat Dragon indicaram a necessidade de uma ferramenta mais madura para a diagramação do modelo. Concorda-se com os achados de Tran, Yskout e Joosen (2023) indicando a importância de estudos sobre usabilidade e curva de aprendizado na modelagem de ameaças em ambientes reais da indústria.

Para o construto 'Intenção de Uso', o ranking médio obtido foi de 4,44, indicando uma proximidade com o Concordo Parcialmente. A variável "Melhoria" indicou que os participantes percebem claramente a melhoria no processo de modelagem de ameaças ao utilizar a ferramenta com o sistema de recomendação proposto. Ao mesmo tempo, a variável "Gostar do Sistema" indicou certa dificuldade e divergência nas respostas sobre a ferramenta. Ressalta-se que os participantes são de diversas áreas de segurança da informação da Organização X, e que em algumas áreas, como, por exemplo, monitoramento de segurança, a modelagem de ameaças não é uma atividade comum no dia a dia, sendo utilizada apenas eventualmente, como na leitura de modelos de ameaças já prontos.

Por fim, considerando o ajuste da tecnologia-tarefa, obteve-se o valor de (4,18), sendo a variável de "Localização das informações" a menor (3,62) entre as cinco variáveis, indicando a dificuldade dos participantes em localizarem as informações facilmente e rapidamente para a execução das atividades de modelagem de ameaças. Esta dificuldade também foi percebida nos demais construtos e nas questões abertas sobre a opinião da ferramenta e das oportunidades de melhoria, estando diretamente relacionada com a usabilidade da ferramenta para execução da tarefa. Quanto às demais variáveis, "Detalhe das Informações", "Atualidade das Informações", "Compreensão" e "Confiabilidade", estas tiveram seus rankings médios com os valores 4,38, 4,08, 4,46 e 4,38, respectivamente. Em conjunto, estas variáveis tiveram 66% das respostas entre "Concordo Parcialmente" e "Concordo Integralmente", indicando que a ferramenta está ajustada para o uso da atividade, embora precise de algumas melhorias. Essas melhorias, segundo os participantes, estão relacionadas com melhor suporte ao uso dos termos e ao aprimoramento das ameaças e do relatório. Entende-se que o melhor suporte nos termos seria tornar facilmente acessível a lista de termos nas ferramentas de diagramação de fluxos de dados e na explicabilidade da decisão de cada termo. Para o aprimoramento das ameaças, espera-se que elas sejam mais específicas e que os relatórios finais possuam o diagrama de fluxo de dados, como forma de documentação. Esta última necessidade também foi observada no trabalho de Tran, Yskout e Joosen (2023).

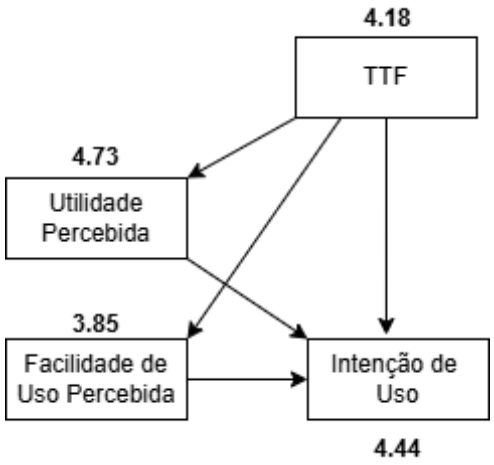
A Figura 43 apresenta os valores quantitativos obtidos para os quatro construtos observados pelo modelo TAM-TTF, bem como as influências existentes entre eles (BOBSIN, 2007). A utilidade percebida influencia tanto a intenção de uso, a facilidade de uso influencia a intenção de uso. E por fim, o ajuste da tecnologia a tarefa influencia a utilidade percebida, a facilidade de uso e a intenção de uso.

Considerando os resultados obtidos, os construtos "Utilidade Percebida" e "Intenção de Uso" apresentaram os maiores valores, sugerindo uma percepção positiva por parte dos participantes quanto à relevância e à aplicabilidade da ferramenta. O ajuste tecnologia-tarefa também obteve um valor elevado, reforçando a importância da adequação da tecnologia às exigências da tarefa. Já o construto "Facilidade de Uso", embora tenha apontado algumas dificuldades, ele não impactou de forma significativa na intenção de uso da ferramenta. O que indica que a "Utilidade Percebida" e o "Ajuste Tecnologia-Tarefa" exerceram influência mais forte na decisão dos usuários em adotar a ferramenta.

Em síntese, os resultados apresentados na Figura 43 reforçam a perspectiva de adoção positiva da ferramenta proposta, destacando que sua utilidade percebida e adequação às atividades de modelagem de ameaças são fatores determinantes para sua aceitação na Organização X avaliada.

Uma observação não prevista inicialmente, mas relevante, é que, durante a execução do exercício de modelagem de ameaças com o uso da ferramenta *Threat Copilot*, a partir da descrição de um sistema exemplo apresentada no Apêndice E, foram observadas diferenças nos diagramas gerados pelos participantes. Cada participante elaborou um DFD com quantidades distintas de atores, processos, armazenamentos e fluxos de dados. Além disso, embora o uso de um vocabulário de termos contribua para reduzir a variabilidade na rotulação, ainda assim foram identificadas diferenças entre os modelos produzidos. Considerando que métodos de modelagem de ameaças baseados em DFD realizam a análise a partir dos elementos e dos fluxos de dados, essas variações podem levar à geração de listas de ameaças diferentes. Assim, tendo em vista que Yskout et al. (2020) apontam a sistematização como um critério importante para a modelagem de ameaças, torna-se relevante que tanto a criação do DFD quanto a rotulação das características dos

Figura 43 – Resultados dos Construtos do TAM-TTF com pontuação



Fonte: O Autor.

elementos e fluxos de dados também sejam conduzidas de forma sistemática. Isso pode contribuir para melhorar a qualidade dos modelos, reduzir a variabilidade entre análises e aumentar a probabilidade de que diferentes pessoas, ao modelarem o mesmo sistema, obtenham resultados equivalentes em termos de ameaças identificadas.

5.4 Discussão da Avaliação Empírica Online de Utilização Real da Ferramenta

No contexto da avaliação empírica online os resultados obtidos, apresentados na Tabela 28 foram 60%, 75% e 67%, respectivamente. Estes resultados foram superiores aos encontrados na avaliação *offline*.

Uma das desvantagens da abordagem *offline* que explica essa diferença de resultados está na limitada disponibilidade de dados e possíveis problemas de qualidade dos dados, como uma representação enviesada do mundo real (GARCÍA, 2024). As métricas coletadas de forma *online* foram feitas em um contexto com diversos participantes, inclusive vários analistas de segurança, representando diversos pontos de vista, resultando em um modelo mais completo, considerando uma maior quantidade de ameaças.

Ao comparar os resultados da avaliação online com aqueles reportados por Tuma et al. (2020), referentes à modelagem de ameaças manual baseada em STRIDE por interação, que apresentaram valores de 60%, 49% e 54% para *precision*, *recall* e *f1-measure*, respectivamente, observa-se um desempenho superior no *recall* e *f1-measure* obtidas pela abordagem proposta nesta pesquisa.

Mesmo superiores, os resultados da avaliação empírica *online* ainda sugerem oportunidades de melhoria no sistema de recomendação da ferramenta *Threat Copilot*. Este achado não difere da dificuldade relacionada com recomendações de ameaças irrelevantes relatada no estudo da adoção da modelagem de ameaças em empresas da Noruega (BERNSMED et al., 2022), quando utilizando o recomendador automático da ferramenta *Microsoft Threat Modeling Tool* (Microsoft, 2018).

Quando são observados os modelos de ameaça da base histórica de conhecimento, percebemos

que muitos destes modelos possuem uma quantidade total de ameaças muito inferior ao total de ameaças recomendado no modelo de ameaças da avaliação empírica. Para exemplificar, o modelo de ameaças médio do conjunto de dados da base de conhecimento possui 16 ameaças para 8 fluxos de dados, enquanto na da avaliação empírica online apresentou ao final 64 ameaças para 11 fluxos de dados. Esta diferença sugere que é possível que haja melhores resultados das métricas de avaliação com novos modelos de ameaça mais completos. Os participantes das questões abertas também indicaram a importância de considerar a data de produção do modelo de ameaças, sugerindo que modelos mais antigos sejam descartados, à medida que os novos sejam elaborados.

Este cenário sugere que o uso da ferramenta *Threat Copilot* com sua base de conhecimento permite uma maior e melhor identificação de ameaças comparados aos modelos de ameaças manuais da Organização X avaliada, fornecendo modelos de ameaças mais completos, e até mesmo uma melhor sistematização na seleção de ameaças. No referido cenário, ao se ter uma lista de ameaças padronizada que deve sempre ser observada por todos os participantes na avaliação dos fluxos de dados do modelo de ameaças, diferentes analistas de segurança consideram a mesma quantidade de ameaças possíveis.

Os participantes também perceberam a importância da recomendação automática e da base de conhecimento de ameaças, já que quando questionados sobre o que acharam da ferramenta, ressaltaram a importância da base de conhecimento de ameaças, e quando questionados sobre as funcionalidades mais importantes da ferramenta, consideraram a recomendação automática de ameaças e a padronização via base de conhecimento de ameaça.

Considerando os critérios de sucesso definidos por Yskout et al. (2020), ser baseado em modelos, rastreável, sistemático, integração com o negócio, suporte a contexto e escalável, estes achados indicam que a ferramenta *Threat Copilot* atende os critérios de sistematização da modelagem de ameaças, uma vez que a elicitação das ameaças é repetível e com um escopo de ameaças claramente definido, de integração ao negócio, já que a base de ameaças foi criada a partir de modelos reais e seus resultados são gerenciados pelos donos dos produtos na organização e de suporte ao contexto, já que as ameaças, seus controles de segurança são oriundos da base de conhecimento são da Organização X avaliada.

Na avaliação online, identificou-se a necessidade de incluir três novas ameaças, o que indica a ocorrência de falsos negativos não apenas entre as recomendações possíveis, mas também em relação a ameaças até então inexistentes na base histórica. Nesses cenários, em que determinadas ameaças não são identificadas automaticamente ou ainda não estão representadas na base de conhecimento, confirma-se que a ferramenta atua como apoio parcial à elicitação, e não como substituta integral do julgamento do especialista, o que torna necessária a revisão humana das recomendações.

Nesse contexto, a principal estratégia de mitigação consiste na curadoria manual das ameaças não recomendadas e na incorporação dessas novas ameaças à base de conhecimento, possibilitando sua recomendação em usos futuros. O caso das ameaças relacionadas à privacidade, ausentes da base histórica e posteriormente identificadas, evidencia esse comportamento incremental e demonstra que a cobertura do sistema depende diretamente da evolução contínua do repositório organizacional de modelos de ameaças. Além disso, a ampliação da capacidade de recomendação pode ser favorecida pela incorporação de modelos de referência e pela inclusão de novos termos no vocabulário, capazes de representar domínios regulatórios, tecnológicos e arquiteturais emergentes, como LGPD, proteção de dados, computação em

nuvem, inteligência artificial e sistemas biométricos.

Embora a ferramenta *Threat Copilot* possibilite a inclusão de novas ameaças, a presente pesquisa não incorporou explicitamente, em seu design, o conceito de novidade. De acordo com Aggarwal (2016), a novidade em um sistema de recomendação corresponde à probabilidade de o sistema fornecer ao usuário recomendações das quais ele ainda não tenha conhecimento ou que não tenha visto anteriormente. Nesse sentido, a incorporação do aspecto temporal das ameaças como critério adicional de ranqueamento pode favorecer a recomendação de ameaças mais recentes, partindo da premissa de que modelos de ameaças mais novos tendem a representar com maior fidelidade a realidade contemporânea.

Ainda no contexto da avaliação empírica, para a indústria de software o tempo de entrega para o mercado é um dos fatores cruciais considerados durante o desenvolvimento de software. Demonstrar que a modelagem de ameaças não possui um impacto no tempo excessivo pode indicar a efetividade e adoção da modelagem de ameaças pelos times de desenvolvimento (TRAN; YSKOUT; JOOSEN, 2023).

Neste contexto, Yskout et al. (2020) definiu estimativas de tempo para atividades relacionadas à modelagem de ameaças, sendo aproximadamente 3 horas para a reunião de *kickoff*, 15 horas para a definição do modelo de ameaças, 16 horas para a elicitação das ameaças e 11 horas para sua finalização. Nesta última etapa, desconsiderou-se o tempo da atividade de entrega para o time de qualidade (4h) por não existir esta atividade na Organização X avaliada, totalizando cerca de 41 horas de trabalho.

Durante a avaliação empírica, foram registrados os tempos de execução de cada atividade para fins de comparação. A reunião de *kickoff* demandou 4 minutos, onde foram apresentados os participantes e o escopo da reunião. A sessão de definição do modelo de ameaças consumiu 2h24min, incluindo a modelagem do DFD, a conversão do modelo a partir da ferramenta *OWASP Threat Dragon* e a revisão do diagrama. A etapa de elicitação de ameaças levou 2h17min, considerando a execução automática das recomendações, a análise e revisão dos resultados, a inclusão de novas ameaças e a revisão dos controles de segurança associados. Por fim, o relatório foi entregue ao dono do produto e aos desenvolvedores em uma reunião de 1 hora. O tempo total utilizado foi de 5h45min.

Resguardadas as diferenças de escopo, tamanho do sistema avaliado, experiência técnica prévia dos participantes, complexidade, requisitos regulatórios, domínio de aplicação e ferramental envolvido, a modelagem de ameaças utilizando a ferramenta *Threat Copilot* utilizou apenas 14% do tempo esperado definido por Yskout et al. (2020).

No contexto do trabalho de Tuma et al. (2021) foram avaliadas duas organizações (A e B) realizando a modelagem de ameaças utilizando o método de modelagem de ameaças STRIDE de forma manual. Neste estudo, foram considerados os tempos de elaboração do DFD, elicitação das ameaças do modelo e priorização, e uma métrica de produtividade baseada na quantidade de ameaças identificadas como verdadeiras por tempo de modelagem de ameaças. Resguardadas as diferenças já apresentadas, com a adicional do domínio dos sistemas da organização A e B serem da área automotiva, a Tabela 30 apresenta um comparativo da Organização X, que é a organização avaliada nesta pesquisa, com as organizações A e B.

Os positivos verdadeiros indicam a quantidade de ameaças identificadas que realmente foram consideradas ameaças reais. A Organização X apresentou uma quantidade superior, mas próxima dos

Tabela 30 – Quadro Comparativo dos Tempos das Organizações

	Organização A	**Organização B**	**Organização X**
Positivos Verdadeiros	11	40	46
Elaboração do DFD	1h10min	1h55min	2h24min
Elicitação das Ameaças	2h20min	4h15min	2h17min
Priorização	0h20min	0h50min	–
Total	3h50min	7h00min	4h41min
Produtividade (A./h)	3,0	5,7	9,8

Fonte: O Autor.

resultados da Organização B e muito superior a Organização A. Os resultados entre a Organização A, B e X sobre a elaboração do DFD quando comparados com o valor de 15h definidos por Yskout et al. (2020) são similares, o da elicitação das ameaças a Organização X usando a ferramenta foi ligeiramente mais ágil do que a organização B, e a priorização das ameaças é feita automaticamente na ferramenta *Threat Copilot*. Considerando a métrica de produtividade, a ferramenta desta pesquisa na Organização X apresentou um desempenho 70% superior a Organização A e 42% superior a Organização B, ambas usando métodos manuais.

Este achado indica que o uso da ferramenta aumenta a produtividade do analista de segurança, sendo um achado também corroborado pelo valor (4,73) do construto "Utilidade Percebida" na avaliação TAM-TTF, onde 75% dos participantes concordaram integralmente com a utilidade da ferramenta e pelas observações positivas sobre a "utilidade" e "valor agregado" da ferramenta *Threat Copilot* conforme apresentado na seção 4.2.3

Um aspecto prático relevante observado na avaliação empírica online foi o tempo necessário para a construção prévia do DFD, etapa indispensável para o uso da *Threat Copilot*. Embora a ferramenta tenha sido proposta para reduzir o esforço na elicitação de ameaças, sua aplicação depende da existência de um modelo inicial do sistema, cuja elaboração demanda tempo adicional por parte da equipe. Em cenários nos quais o DFD ainda não está disponível, esse requisito pode reduzir a percepção de ganho imediato no uso da ferramenta. Esta percepção pode ser corroborada quando um dos participantes do estudo dos usuários com TAM-TTF informou sobre a necessidade de ter pessoas experientes para a realização da modelagem de ameaças.

Essa limitação torna-se mais evidente quando se considera que o tempo investido não se restringe ao desenho do diagrama, mas também envolve a identificação adequada dos elementos, dos fluxos de dados e de sua rotulação. Como diferentes participantes podem representar o mesmo sistema de formas distintas, o tempo total da atividade também pode variar, comprometendo a sistematização do processo, a qualidade do modelo e gerando modelos de ameaças variáveis. Naturalmente esta variabilidade afetará a métrica de produtividade de ameaças por hora.

Assim, do ponto de vista prático, a viabilidade da *Threat Copilot* está condicionada não apenas ao tempo economizado na recomendação de ameaças, mas também ao tempo previamente necessário para estruturar o DFD. Esse fator representa uma limitação importante da abordagem, especialmente em organizações com baixa maturidade em modelagem de ameaças ou com pouca padronização na representação arquitetural dos sistemas.

6 CONCLUSÃO

Neste capítulo são apresentadas as considerações finais desta pesquisa, consolidando as principais contribuições obtidas. Além disso, serão apresentadas as publicações obtidas e propostas para trabalhos futuros, identificando áreas, potenciais melhorias metodológicas e possíveis aplicações práticas dos resultados alcançados. Por fim, também serão apresentadas as limitações deste estudo realizado e as ameaças a sua validade.

6.1 Resumo das Contribuições

Esta pesquisa apresentou o *Threat Copilot*, um sistema de recomendação para apoiar a automação da elicitação de ameaças em processos de modelagem de ameaças de software. A proposta surgiu do desafio reconhecido na literatura e na prática: a modelagem de ameaças ainda demanda tempo significativo, elevado nível de expertise e baixa disponibilidade de conhecimento prévio estruturado nas organizações. Adicionalmente, Yskout et al. (2020) também mencionou o baixo nível de maturidade, nos termos de pesquisa, suporte de ferramentas e abordagens práticas.

Diante desse cenário, foi desenvolvida uma abordagem que reutiliza modelos de ameaças existentes para recomendar ameaças relevantes a novas arquiteturas de sistemas, contribuindo para maior celeridade e consistência no processo de desenvolvimento seguro.

A partir da estruturação de um dataset com modelos reais de uma organização pública de tecnologia da informação, da definição de uma unidade de ameaça e do emprego de métricas de similaridade com pesos, combinadas a um mecanismo de ranqueamento, o *Threat Copilot* foi capaz de identificar ameaças com as métricas *offline* de *precision*, *recall* e *f1-measure* e valores de 51%, 72% e 56%, respectivamente.

Além da análise das métricas do sistema de recomendação, uma avaliação centrada no usuário foi conduzida com especialistas da Organização X, por meio do modelo TAM-TTF, permitindo analisar "facilidade de uso percebida", "utilidade percebida", "ajuste tarefa-tecnologia" e "intenção de uso" do sistema. Respectivamente as médias dos resultados de 3.85, 4.73, 4.44 e 4.18, indicaram percepção de concordar quanto ao valor oferecido pela ferramenta, especialmente no ganho de produtividade e no apoio à execução da modelagem de ameaças

Adicionalmente, foi realizada uma avaliação online real utilizando a ferramenta. A avaliação *online* empírica em um cenário real demonstrou desempenho superior nas métricas *precision* (9%), *recall* (3%) e *f1-measure* (11%) aos obtidos nos testes offline, mas com uma quantidade superior de ameaças identificadas (64) em relação aos 33 dos 34 modelos de ameaças da base de conhecimento, sendo que esses possuem até 45 ameaças. Essa diferença sugeriu que os novos modelos de ameaças serão mais completos e quando inseridos na base de conhecimento, impactarão diretamente a precisão das recomendações futuras. O estudo demonstrou que a ferramenta não apenas sistematiza a identificação de ameaças, mas também eleva a quantidade de ameaças identificadas, fornecendo uma base de conhecimento padronizada e evolutiva para a organização.

Esta pesquisa sugere que a automação da modelagem de ameaças, baseada em um sistema de recomendação a partir de modelos de ameaças pré-existentes, pode acelerar a prática, promover o reuso de conhecimento organizacional e ampliar a maturidade da prática de modelagem de ameaças nas instituições que adotam a modelagem de ameaças como parte do seu processo de desenvolvimento seguro. Ao mesmo tempo, contribui para o avanço científico ao introduzir um modelo de recomendação voltado especificamente ao domínio da segurança de software, baseado em dados reais de uma empresa pública de TI.

A presente dissertação apresenta contribuições relevantes para três dimensões fundamentais: teórica, metodológica e aplicada. As próximas subseções detalham estas contribuições.

6.1.1 Contribuição Teórica

O estudo aprofunda o conhecimento no campo de desenvolvimento seguro ao propor uma abordagem inovadora que integra sistemas de recomendação ao processo de modelagem de ameaças. O estudo contribui para o corpo teórico ao definir o conceito de unidade de ameaça como elemento comparável para a recomendação, estruturado a partir das relações entre os componentes de um diagrama de fluxo de dados. Essa definição contribui com uma lacuna na literatura, que ainda carece de modelos formais para representação e reuso de conhecimento em modelagem de ameaças. Ao integrar métodos de avaliação de aceitação tecnológica (TAM-TTF) ao domínio da modelagem de ameaças, este estudo também fortalece a compreensão teórica sobre os fatores que influenciam a adoção de ferramentas de automação por profissionais de segurança da informação.

Como parte do desenvolvimento desta pesquisa, foram publicados dois artigos, um que será apresentado em maio de 2026 no 28th International Conference on Enterprise Information Systems (ICEIS) e o outro publicado em setembro de 2023 um artigo científico no Salão de Ferramentas do XXIII Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais (SBSEG). Adicionalmente, também foi publicado um registro de software para a versão desenvolvida na época.

- NEGÓCIO, Y. F.; V. DE MEDEIROS, J. D. R. RODRIGUES, N. N. A Recommender System for Automated Security Threat Elicitation in Enterprise Information Systems. 28th International Conference on Enterprise Information Systems (ICEIS). 2026.
- NEGOCIO, Y. F., JUNIOR, N. T. H., MEDEIROS, J. D. R V. Threat Copilot: Um sistema de recomendação para a modelagem de ameaças. Salão de Ferramentas XXIII Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais. 2023. Juiz de Fora. Brasil.
- PESSOA, D. E. R. ; FERNANDES, D. Y. S. ; NEGOCIO, Y. F. . Sistema de Apoio a Modelagem de Ameaças de Software. 2023. Patente: Programa de Computador. Número do registro: BR512023002511-9, data de registro: 01/03/2023, título: "Sistema de Apoio a Modelagem de Ameaças de Software", Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial

6.1.2 Contribuição Metodológica

A pesquisa adota um delineamento quantitativo, qualitativo e descritivo integrando duas perspectivas complementares de avaliação. Primeiramente, o desempenho do sistema de recomendação foi

analisado por meio de métricas comuns aos sistemas de recomendação (*precision*, *recall* e *f1-measure*), obtidas por meio de experimentos *offline* e reforçadas por uma avaliação empírica *online* em cenário real de uso. Em paralelo, foi conduzida uma investigação *online* centrada no usuário fundamentada na integração dos modelos TAM e TTF, permitindo compreender fatores que influenciam a adoção tecnológica, como utilidade percebida, facilidade de uso, intenção de uso e ajuste tarefa-tecnologia. O instrumento adotado foi elaborado com base em literatura consolidada e validado em uma execução de modelagem de ameaças real, o que sugeriu maior confiabilidade e robustez ao processo de coleta e análise dos dados.

6.1.3 Contribuição aplicada

Os resultados oferecem diretrizes práticas para empresas públicas interessadas em aplicar a automação da modelagem de ameaças nos seus processos de desenvolvimento seguro de software. Adicionalmente, temos as seguintes contribuições:

- a linguagem de descrição de modelos de ameaças (UTML);
- um dataset detalhado dos modelos de ameaças efetivamente utilizados;
- a definição do conceito de unidade de ameaça, que representa a base de comparação para o cálculo de similaridade;
- os algoritmos de similaridade e a definição dos pesos por consenso de especialistas aplicados nesse cálculo;
- a construção de uma base de conhecimento de ameaças e controles;
- a elaboração de um vocabulário (WordNet) estruturado hierarquicamente de propósito específico dos termos relacionados aos modelos de ameaças;
- uma linguagem simplificada para a construção de WordNet;
- as métricas de *precision*, *recall* e *f1-measure* do sistema de recomendação de ameaças *offline* e *online*;
- a ferramenta do sistema de recomendação *Threat Copilot*.

Há potencial de generalização dos resultados desta pesquisa para outras organizações que compartilham características semelhantes às analisadas. Embora o estudo tenha sido conduzido em um ambiente específico, os conceitos, métodos e artefatos desenvolvidos podem ser adaptados e aplicados a empresas que enfrentam desafios equivalentes.

6.2 Limitações e Ameaças à Validade

Este estudo apresenta algumas limitações que devem ser consideradas na interpretação dos resultados e na generalização dos achados, sob a ótica dos tipos de ameaças definidas por Juristo e Moreno (2001).

6.2.1 Ameaças a validade de conclusão

As ameaças à validade de conclusão referem-se a fatores que podem comprometer a interpretação dos resultados e dificultar a obtenção de conclusões confiáveis a partir dos dados (JURISTO; MORENO, 2001). O fato da pesquisa ter sido conduzida em uma única organização reduz o poder de generalização estatística dos resultados. Como os achados refletem apenas o contexto analisado, outras realidades organizacionais podem apresentar comportamentos distintos, limitando a confiabilidade das inferências obtidas. Na avaliação centrada no usuário, um número reduzido de participantes (13 profissionais) limita o poder estatístico do estudo e impede análises quantitativas mais robustas, permitindo apenas conclusões qualitativas e descritivas. Contudo, ressalta-se que todos os participantes se encaixavam como público-alvo da ferramenta (profissionais de segurança da informação com experiência em modelagem de ameaças), o que contribui positivamente para a qualidade e relevância das percepções coletadas.

6.2.2 Ameaças a validade de construção

As ameaças à validade de construção estão relacionadas com o desenho do estudo para garantir se o que foi medido simula as condições reais do que deseja avaliar (JURISTO; MORENO, 2001). O tamanho e a qualidade da base de conhecimento de ameaças devem ser considerados nos resultados. Os modelos de ameaças foram gerados por profissionais com diversos níveis de experiência profissional, mas em alguns casos, os modelos de ameaças consideraram apenas uma lista limitada de ameaças. Nesta pesquisa foi utilizada uma única técnica de recomendação de ameaças baseada no ranqueamento top-N. Os resultados podem refletir apenas o desempenho desse algoritmo específico, sem representar adequadamente o potencial da abordagem de recomendação de ameaças como um todo. Isso limita a interpretação dos achados, que passam a indicar apenas o funcionamento do top-N no contexto estudado, e não a eficácia geral da solução proposta em diferentes cenários ou com outros métodos e algoritmos.

6.2.3 Ameaças à validade interna

As ameaças à validade interna estão relacionadas ao que pode afetar a variável independente do experimento (JURISTO; MORENO, 2001).

A coleta e a classificação dos modelos de ameaça foram realizadas exclusivamente pelo autor, o que pode introduzir viés do pesquisador durante a interpretação dos resultados. Esse aspecto representa uma ameaça à validade interna, uma vez que decisões subjetivas podem influenciar a relação entre o uso da ferramenta e os efeitos observados.

6.2.4 Ameaças à validade externa

As ameaças à validade externa são aquelas que estão relacionadas com a generalização dos resultados em um ambiente externo (JURISTO; MORENO, 2001). A pesquisa foi conduzida considerando um conjunto de modelos de ameaças elaborados em 2019 e com profissionais de segurança da informação que atuaram na avaliação em 2025. Mudanças institucionais, evolução dos processos e maior maturidade na adoção da modelagem de ameaças podem alterar o comportamento dos usuários e os artefatos produzidos, impactando os resultados.

A amostragem adotada para a seleção dos participantes na definição dos pesos e na avaliação centrada do usuário pode ter introduzido viés de seleção, uma vez que os profissionais escolhidos ocupam posições estratégicas e possuem envolvimento direto com o tema de segurança da informação. Isso pode ter favorecido a obtenção de percepções mais estruturadas orientadas para este tipo de profissional, em detrimento de visões mais críticas ou divergentes que poderiam ser realizadas por outros perfis profissionais do desenvolvimento de software.

Também pode ter havido viés de resposta, decorrente da tendência dos entrevistados em relatar experiências positivas ou omitir fragilidades, especialmente em ambientes nos quais há receio de exposição institucional ou pessoal.

A avaliação empírica *online* foi restrita à observação de um único cenário operacional (5 profissionais de segurança da informação, 3 arquitetos de software e 1 desenvolvedor). Embora essa abordagem tenha fornecido métricas de *precision*, *recall* e *f1-measure* em um contexto de uso real, ela restringe a representatividade e a diversidade do conjunto de dados, limitando a capacidade de generalizar as conclusões sobre a eficácia do sistema de recomendação para uma população mais ampla de usuários ou para diferentes contextos de modelagem de ameaças.

As métricas de avaliação do sistema de recomendação são altamente sensíveis a WordNet de domínio específico utilizado. A restrição dos dados coletados ao uso particular deste vocabulário específico limita a generalização do desempenho para cenários com outras WordNets.

Por fim, outra limitação refere-se ao recorte institucional da pesquisa, restrita a uma única empresa pública de tecnologia da informação. Esse recorte também impõe restrições à generalização dos resultados, considerando que práticas, maturidade institucional e contextos organizacionais podem variar significativamente entre diferentes empresas.

6.2.5 Conclusões

Essas limitações e ameaças, reconhecidas e explicitadas, não comprometem a relevância do estudo, mas reforçam o compromisso com a transparência científica e com a utilidade prática da pesquisa. Apesar dessas limitações, os resultados obtidos sugerem indícios relevantes sobre o potencial do sistema de recomendação proposto, justificando sua evolução e a realização de estudos adicionais, com maior controle experimental e variedade de casos, visando reduzir as ameaças à validade identificadas.

6.3 Trabalhos Futuros

Com relação aos trabalhos futuros, com base nas limitações e potencialidades identificadas, foram destacadas algumas propostas possíveis de continuidade e de aprofundamento na pesquisa. Elas são listadas a seguir:

- Os modelos de ameaças são constituídos muitas vezes com informações sigilosas das organizações. Para que datasets maiores sejam construídos com a contribuição de diversas organizações, deve ser possível armazenar e compartilhar os modelos de forma que não representem riscos à segurança da informação da organização.

- A quantidade de modelos identificados nesta pesquisa demonstra a dificuldade de ter acesso a uma quantidade real significativa de modelos, assim como também modelos completos e de qualidade. Para evolução geral do tema de pesquisa, deve-se criar esforços para identificar e enriquecer uma maior quantidade de modelos.

- O uso de modelos de ameaças na forma de relatórios não estruturados é uma dificuldade na agilidade da criação dos datasets. Pesquisas que buscam propor transformações de modelos de ameaça não estruturados em modelos de ameaça estruturados facilitam o aumento do dataset de modelos de ameaça.

- O conceito inicial de unidade de ameaça (*threat unit*) definido nesta pesquisa pode ser expandido para a criação de novos atributos de termos, podendo, dessa forma, representar com mais detalhes as características dos elementos que compõem o modelo de ameaça. Espera-se que, com novos atributos, melhore-se a acurácia das recomendações.

- A função de definição de similaridade pode ser aprimorada por meio da escolha dos algoritmos de similaridade para nomes compostos e da comparação dos termos na taxonomia. Estudos comparando os diversos algoritmos e verificando a evolução das métricas podem melhorar a acurácia das recomendações.

- A definição dos pesos utilizados na função de similaridade pode ser evoluída através de outras técnicas quantitativas ou qualitativas. Um balanceamento dos pesos, de forma mais precisa, pode representar melhores métricas nas recomendações.

- A taxonomia inicial também pode ser evoluída. Estudos que possam identificar identificando melhor as relações hierárquicas de hiperônimos, hipônimos e homônimos entre os termos, e possam identificar ainda quais conceitos são comuns aos modelos de ameaça, podem melhorar a abrangência dos termos possíveis nos modelos e naturalmente melhorar as métricas das recomendações.

- Critérios temporais das ameaças podem ser integrados ao ranqueamento das recomendações, permitindo maior relevância para ameaças associadas a modelos mais recentes e, portanto, mais aderentes ao contexto tecnológico, arquitetural e regulatório atual da organização.

- Os diagramas de fluxo de dados que pertencem aos modelos de ameaça podem ser construídos de diversas formas, o que naturalmente gera modelos e identificações de ameaças distintas. Pesquisas voltadas a identificação de padrões e boas práticas na definição dos diagramas e dos elementos necessários auxiliarão na elaboração de modelos mais completos.

- Recentemente, estudos investigaram o uso de large language models (LLMs) como apoio à modelagem de ameaças, explorando seu potencial para automatizar etapas desse processo. Novas pesquisas podem analisar a viabilidade e os resultados obtidos.

Em resumo, conclui-se que a pesquisa aqui apresentada não se configura como um encerramento do estudo, e sim um ponto de partida para que novos trabalhos futuros possam contribuir de maneira significativa e continuar contribuindo para a geração de melhores modelos de ameaça, principalmente, de

forma mais ágil e sistemática. Espera-se que a evolução no tema permita que os times de desenvolvimento de software adotem cada vez mais a modelagem de ameaças.

REFERÊNCIAS BIBLIOGRÁFICAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, v. 17, n. 6, p. 734–749, 2005. Citado na página 69.

AGGARWAL, C. C. *Recommender Systems*. Cham: Springer International Publishing, 2016. Citado 7 vezes nas páginas 28, 29, 49, 68, 75, 88 e 112.

AL-MATOUQ, H. et al. A maturity model for secure software design: A multivocal study. *IEEE Access*, v. 8, p. 215758–215776, 2020. Citado na página 23.

ASSEN, J. von der et al. Coretm: An approach enabling cross-functional collaborative threat modeling. In: *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*. [S.l.: s.n.], 2022. Citado na página 32.

BANU, A.; FATIMA, S. S.; KHAN, K. U. R. A survey and comparison of wordnet based semantic similarity measures. *International Journal of Computer Science And Technology*, v. 42, n. 3, p. 456–461, 2013. Citado 2 vezes nas páginas 71 e 72.

BEHERA, P. C.; DASH, C.; PAREEK, P. K. A novel approach for improving security in software development in small software firms: A literature review. In: TAVARES, J. M. R. S. et al. (Ed.). *Emerging Technologies in Data Mining and Information Security*. Singapore: Springer Singapore, 2021. Citado na página 17.

BERNSMED, K. et al. Adopting threat modelling in agile software development projects. *Elsevier*, 2022. Citado 3 vezes nas páginas 17, 108 e 110.

BeyondTrust. *2023 Microsoft Vulnerabilities Report*. 2023. Acesso em: 12 jul. 2023. Disponível em: <https://www.beyondtrust.com/resources/whitepapers/microsoft-vulnerability-report>. Citado na página 17.

BOBSIN, D. *A percepção dos diferentes níveis hierárquicos quanto ao uso de um sistema de informação*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2007. Citado na página 109.

BOND, F. et al. Some issues with building a multilingual wordnet. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. [S.l.: s.n.], 2020. p. 3189–3197. Citado na página 58.

Brasil. *Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)*. 2018. Diário Oficial da União, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 12 out. 2025. Citado na página 102.

CARDOSO, L.; CLEOPHAS, M. das G. Avaliação do uso dos dispositivos móveis no ensino de química por meio da adaptação dos modelos de aceitação (tam) e ajuste (ttf). *Debates em Educação*, v. 12, n. 27, p. 818–840, 2020. Citado 3 vezes nas páginas 88, 95 e 108.

CASOLA, V. et al. Toward the automation of threat modeling and risk assessment in iot systems. *Internet of Things*, v. 7, 2019. Citado na página 17.

CHANDRASEKARAN, D.; MAGO, V. Evolution of semantic similarity—a survey. *ACM Computing Surveys*, v. 54, n. 2, p. 1–37, 2021. Citado na página 72.

CHO, H.; KIM, S. Threat modeling for the defense industry: Past, present, and future. *IEEE Access*, v. 13, p. 135208–135223, 2025. Citado 2 vezes nas páginas 17 e 24.

DISHAW, M. T.; STRONG, D. M. Extending the technology acceptance model with task–technology fit constructs. *Information & Management*, v. 36, n. 1, p. 9–21, 1999. Citado 2 vezes nas páginas 21 e 107.

ELIYAHU, T. *Threat Model Examples*. 2022. Acesso em: 10 jan. 2024. Disponível em: <https://github.com/TalEliyahu/Threat_Model_Examples>. Citado na página 51.

ELKAMEL, A.; GZARA, M.; BEN-ABDALLHAH, H. An uml class recommender system for software design. In: *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. [S.l.: s.n.], 2016. Citado na página 71.

ERIKSSON, M.; HALLBERG, V. *Comparison between JSON and YAML for Data Serialization*. [S.l.], 2011. Citado na página 54.

ERL, T.; KHATTAK, W.; BUHLER, P. *Big Data Fundamentals: Concepts, Drivers & Techniques*. [S.l.]: Prentice Hall, Service Tech Press, 2016. Citado na página 52.

FAILY, S. *Designing Usable and Secure Software with IRIS and CAIRIS*. Cham: Springer International Publishing, 2018. Citado na página 32.

FERREIRA, L.; SILVA, D. C.; URIARTE, M. Recommender systems in cybersecurity. *Knowledge and Information Systems*, v. 65, n. 12, p. 1–37, 2023. Citado 2 vezes nas páginas 18 e 49.

FILHO, E. D. S. *Uma Abordagem para Recomendação de Casos de Teste em Projetos Ágeis Baseados em Scrum*. Tese (Doutorado) — Universidade Federal de Campina Grande, Paraíba, Brasil, 2021. Citado 4 vezes nas páginas 30, 64, 80 e 85.

FILHO, M. C. S.; RODRIGUES, J. J. P. C. Human readable scenario specification for automated creation of simulations on cloudsim. In: *International Conference on Internet of Vehicles*. Cham: Springer International Publishing, 2014. Citado na página 54.

FRANCIS, N. et al. Cypher: An evolving query language for property graphs. In: *Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18)*. Houston, EUA: [s.n.], 2018. Citado na página 81.

FRYDMAN, M. et al. Automating risk analysis of software design models. *The Scientific World Journal*, 2014. Citado 4 vezes nas páginas 32, 34, 35 e 75.

GANDOMANI, T. J.; WEI, K. T.; BINHAMID, A. K. A case study research on software cost estimation using experts' estimates, wideband delphi, and planning poker technique. *International Journal of Software Engineering and Its Applications*, v. 8, n. 11, p. 173–182, 2014. Citado na página 75.

GARCÍA, L. A. *Engineering Recommender Systems for Modelling Languages: An Automated End-to-end Approach*. Tese (Doutorado) — Universidad Autónoma de Madrid, Madrid, 2024. Citado 4 vezes nas páginas 85, 100, 106 e 110.

GASIBA, T. E. et al. Is secure coding education in the industry needed? an investigation through a large scale survey. In: *IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. [S.l.: s.n.], 2021. p. 241–252. Citado na página 17.

GASPARIC, M.; JANES, A. What recommendation systems for software engineering recommend: A systematic literature review. *Journal of Systems and Software*, v. 113, p. 101–113, 2016. Citado na página 18.

GHOSH, S. et al. An integrated approach of threat analysis for autonomous vehicles perception system. *IEEE Access*, v. 11, p. 14752–14777, 2023. Citado na página 17.

GOMES, P. J. d. S. *A Case-Based Approach to Software Design*. Tese (Doutorado) — Universidade de Coimbra, 2004. Citado na página 71.

GOODHUE, D. L.; THOMPSON, R. L. Task-technology fit and individual performance. *MIS Quarterly*, v. 19, n. 2, p. 213–236, 1995. Citado na página 21.

GOODMAN, M. W.; BOND, F. Intrinsically interlingual: The wn python library for wordnets. In: *Proceedings of the 11th Global Wordnet Conference*. [S.l.: s.n.], 2021. Citado 2 vezes nas páginas 58 e 81.

GRANATA, D.; RAK, M. Systematic analysis of automated threat modelling techniques: Comparison of open-source tools. *Software Quality Journal*, v. 32, n. 1, p. 125–161, 2023. Citado 6 vezes nas páginas 20, 32, 33, 34, 36 e 48.

GRANATA, D.; RAK, M.; SALZILLO, G. Automated threat modeling approaches: Comparison of open source tools. In: *International Conference on the Quality of Information and Communications Technology (QUATIC 2022)*. Cham: Springer International Publishing, 2022. Citado 2 vezes nas páginas 32 e 47.

GRINBERG, M. *Flask Web Development: Developing Web Applications with Python*. [S.l.]: O'Reilly Media, 2018. Citado na página 81.

GRUBER, J.; SWARTZ, A. *Markdown Language*. 2004. Acesso em: 10 jan. 2024. Disponível em: <https://daringfireball.net/projects/markdown/>. Citado na página 53.

IriusRisk. *Open Threat Modeling Format (OTM)*. 2023. Acesso em: 1 abr. 2024. Disponível em: <https://github.com/iriusrisk/OpenThreatModel/>. Citado na página 54.

JURISTO, N.; MORENO, A. M. *Basics of Software Engineering Experimentation*. [S.l.]: Kluwer Academic Publishers, 2001. Citado 2 vezes nas páginas 116 e 117.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering — a systematic literature review. *Information and Software Technology*, v. 51, n. 1, p. 7–15, 2009. Citado na página 32.

KU Leuven, DistriNet Research Group. *SPARTA – Security & Privacy Architecture Tool for Risk Analysis*. 2022. Acesso em: 1 out. 2025. Disponível em: <https://docs.sparta.distrinet-research.be/main/index.html>. Citado 2 vezes nas páginas 39 e 40.

KUDRIAVTSEVA, A.; GADYATSKAYA, O. *Secure Software Development Methodologies: A Multivocal Literature Review*. 2022. ArXiv preprint. ArXiv:2211.16987. Acesso em: 22 mar. 2023. Disponível em: <http://arxiv.org/abs/2211.16987>. Citado 2 vezes nas páginas 17 e 23.

LANDUYT, D. V. et al. From automation to ci/cd: a comparative evaluation of threat modeling tools. In: IEEE. *2024 IEEE Secure Development Conference (SecDev)*. [S.l.], 2024. p. 35–45. Citado na página 38.

LEACOCK, C.; CHODOROW, M. Combining local context and wordnet similarity for word sense identification. In: FELLBAUM, C. (Ed.). *WordNet: An Electronic Lexical Database*. [S.l.]: MIT Press, 1998. p. 265–283. Citado 3 vezes nas páginas 58, 72 e 87.

LECHNER, N. H.; STRAHONJA, V.; STAPIĆ, Z. *Threat Modeling Methods in the Medical Device Industry: An Integrative Literature Review*. 2022. Citado na página 17.

LINSTONE, H. A.; TUROFF, M. *The Delphi Method: Techniques and Applications*. Reading, Massachusetts: Addison-Wesley, 1975. Citado na página 75.

Markdown-it. *Markdown-it: Markdown Parser Done Right*. 2025. Acesso em: 27 out. 2025. Disponível em: <https://markdown-it.github.io/>. Citado na página 82.

MARTINS, G. et al. *Towards a Systematic Threat Modeling Approach for Cyber-Physical Systems*. Gaithersburg, MD, 2015. Citado na página 32.

MCGRAW, G. *Software Security: Building Security In*. [S.l.]: Addison-Wesley, 2006. Citado na página 23.

MELAND, P. H. et al. Threat analysis in goal-oriented security requirements modelling. *International Journal of Secure Software Engineering*, v. 5, n. 2, p. 1–19, 2014. Citado na página 32.

MENDONÇA, C. V. *Sistemas de Recomendação Científica: Incorporação da Reputação Científica nos seus Algoritmos*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2022. Citado na página 75.

Mermaid. *Mermaid: Diagramming and Charting Tool*. 2025. Acesso em: 27 out. 2025. Disponível em: <https://mermaid.js.org/>. Citado na página 82.

Microsoft. *Microsoft Threat Modeling Tool*. 2018. Acesso em: 8 out. 2023. Disponível em: <https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool>. Citado 2 vezes nas páginas 32 e 110.

Microsoft Corporation. *Microsoft Teams [software]. Versão disponível em 2025*. 2017. Acesso em: 12 out. 2025. Disponível em: <https://www.microsoft.com/microsoft-teams>. Citado 3 vezes nas páginas 77, 90 e 101.

MILLER, G. A. Wordnet: A lexical database for english. *Communications of the ACM*, v. 38, n. 11, p. 39–41, 1995. Citado 2 vezes nas páginas 58 e 71.

Neo Technology. *Neomodel: Object Graph Mapper (OGM) for Neo4j*. 2025. Acesso em: 27 out. 2025. Disponível em: <https://neomodel.readthedocs.io/>. Citado na página 82.

Neo4j Inc. *Neo4j Graph Database Platform*. 2025. Acesso em: 27 out. 2025. Disponível em: <https://neo4j.com/>. Citado na página 81.

NIST. *Guide to Data-Centric System Threat Modeling (NIST Special Publication 800-154)*. Gaithersburg, MD, 2016. Citado na página 51.

OWASP. *OWASP Automated Threats to Web Applications*. 2019. Acesso em: 1 abr. 2024. Disponível em: <https://owasp.org/www-project-automated-threats-to-web-applications/>. Citado na página 38.

OWASP. *OWASP Threat Cookbook*. 2019. Acesso em: 10 jan. 2024. Disponível em: <https://github.com/OWASP/threat-model-cookbook>. Citado na página 51.

OWASP. *PYTM: Pythonic Framework for Threat Modeling*. 2019. Acesso em: 8 ago. 2023. Disponível em: <https://owasp.org/www-project-pytm/>. Citado 4 vezes nas páginas 32, 41, 43 e 44.

OWASP. *OWASP Threat Dragon*. 2020. Acesso em: 8 out. 2023. Disponível em: <https://github.com/OWASP/threat-dragon>. Citado 2 vezes nas páginas 32 e 37.

PAIDY, P. Leveraging ai in threat modeling for enhanced application security. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, v. 4, n. 2, p. 57–66, 2023. Citado na página 51.

PAWLICKA, A. et al. A systematic review of recommender systems and their applications in cybersecurity. *Sensors*, MDPI, v. 21, n. 15, p. 5248, 2021. Citado na página 49.

Python Software Foundation. *Python: A Programming Language*. 2025. Acesso em: 27 out. 2025. Disponível em: <https://www.python.org/>. Citado na página 81.

RADA, R. et al. Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 19, n. 1, p. 17–30, 1989. Citado 2 vezes nas páginas 72 e 87.

RAMOS, F. B. A.; SILVA, J. P. L.; CASTRO, J. M. N. A non-functional requirements recommendation system for scrum-based projects. In: *30th International Conference on Software Engineering and Knowledge Engineering (SEKE 2018)*. Redwood City, CA: KSI Research Inc., 2018. Citado na página 57.

RICCI, F.; ROKACH, L.; SHAPIRA, B. *Recommender Systems Handbook*. Boston, MA: Springer US, 2010. Citado 2 vezes nas páginas 29 e 85.

ROBILLARD, M. P. et al. (Ed.). *Recommendation Systems in Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. Citado 4 vezes nas páginas 18, 30, 80 e 85.

ROCCO, J. D. et al. Development of recommendation systems for software engineering: The crossminer experience. *Empirical Software Engineering*, v. 26, n. 4, p. 69, 2021. Citado 3 vezes nas páginas 18, 30 e 31.

ROCCO, J. D. et al. On the need for a body of knowledge on recommender systems. In: *KaRS / ComplexRec Workshop @ RecSys 2021*. [S.l.: s.n.], 2021. Citado na página 30.

ROSA, F. D. et al. Threma: Ontology-based automated threat modeling for ict infrastructures. *IEEE Access*, v. 10, p. 116514–116526, 2022. Citado na página 18.

ROSZKOWSKA, E. Rank ordering criteria weighting methods–a comparative overview. *Optimum. Studia Ekonomiczne*, n. 5(65), p. 14–33, 2013. Citado 2 vezes nas páginas 76 e 77.

SCANDARIATO, R.; WUYTS, K.; JOOSEN, W. A descriptive study of microsoft's threat modeling technique. *Requirements Engineering*, v. 20, n. 2, p. 163–180, 2015. Citado na página 26.

SCHAAD, A.; BOROZDIN, M. Tam 2: Automated threat analysis. In: *Proceedings of the 27th ACM Symposium on Applied Computing (SAC 2012)*. Trento: [s.n.], 2012. Citado na página 32.

SHI, Z. et al. Threat modeling tools: A taxonomy. *IEEE Security & Privacy*, 2021. Citado 2 vezes nas páginas 36 e 48.

SHIBANI, A. S. M. et al. Identification of critical parameters affecting an e-learning recommendation model using delphi method based on expert validation. *Information*, 2023. Citado na página 75.

SHOSTACK, A. *Threat Modeling: Designing for Security*. [S.l.]: John Wiley & Sons, 2014. Citado 2 vezes nas páginas 18 e 66.

SION, L. et al. Sparta: Security & privacy architecture through risk-driven threat assessment. In: *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*. [S.l.: s.n.], 2018. Citado 4 vezes nas páginas 32, 38, 39 e 41.

SOLMS, R. von; NIEKERK, J. van. From information security to cyber security. *Computers & Security*, v. 38, p. 97–102, 2013. Citado na página 23.

Synopsys. *BSIMM 13 Foundations Report*. 2022. Acesso em: 22 mar. 2023. Disponível em: <https://www.synopsys.com/software-integrity/software-security-services/bsimm-maturity-model.html>. Citado 2 vezes nas páginas 23 e 24.

The Go Authors. *The Go Programming Language*. 2025. Acesso em: 1 out. 2025. Disponível em: <https://go.dev/>. Citado na página 45.

The Open Group. *ArchiMate 3.2 Specification*. Reading, UK: The Open Group, 2022. Citado na página 102.

Threatagile. *Threatagile - Agile Threat Modeling*. 2020. Acesso em: 3 mar. 2024. Disponível em: <https://threagile.io/>. Citado 2 vezes nas páginas 33 e 45.

ThreatSpec. *Threat Spec: Continuous Threat Modeling Through Code*. 2019. Acesso em: 3 mar. 2024. Disponível em: <https://threatspec.org/>. Citado na página 32.

TMS. *Threat Manager Studio*. 2020. Acesso em: 6 mar. 2024. Disponível em: <https://threatsmanager.com/>. Citado 2 vezes nas páginas 32 e 45.

TOK, Y. C.; CHATTOPADHYAY, S. Identifying threats, cybercrime and digital forensic opportunities in smart city infrastructure via threat modeling. *Forensic Science International: Digital Investigation*, v. 45, 2023. Citado na página 17.

TRAN, A.; YSKOUT, K.; JOOSEN, W. Threat modeling: A rough diamond or fool's gold? In: *European Conference on Software Architecture*. [S.l.]: Springer Nature Switzerland, 2023. Citado 3 vezes nas páginas 108, 109 e 112.

TUMA, K.; CALIKLI, G.; SCANDARIATO, R. Threat analysis of software systems: A systematic literature review. *Journal of Systems and Software*, v. 144, p. 275–294, 2018. Citado na página 18.

TUMA, K. et al. Finding security threats that matter: Two industrial case studies. *Elsevier*, 2021. Citado na página 112.

TUMA, K. et al. Automating the early detection of security design flaws. In: *ACM/IEEE 23rd International Conference on Model Driven Engineering Languages and Systems (MODELS'20)*. Virtual Event, Canada: [s.n.], 2020. Citado 3 vezes nas páginas 18, 51 e 110.

VANHOEFER, J. et al. yaml2sbml: Human-readable and -writable specification of ode models and their conversion to sbml. *Journal of Open Source Software*, 2021. Citado na página 54.

WU, T. et al. *ThreatModeling-LLM: Automating Threat Modeling Using Large Language Models for Banking System*. 2025. ArXiv preprint arXiv:2411.17058v2. Citado na página 51.

WU, Z.; PALMER, M. Verb semantics and lexical selection. In: *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*. [S.l.: s.n.], 1994. Citado 9 vezes nas páginas 9, 11, 58, 72, 73, 82, 87, 88 e 106.

XIONG, W.; LAGERSTRÖM, R. Threat modeling–a systematic literature review. *Computers & Security*, v. 84, p. 53–69, 2019. Citado 3 vezes nas páginas 18, 24 e 26.

YAML. *YAML Ain't Markup Language*. 2001. Acesso em: 1 abr. 2024. Disponível em: <https://yaml.org/>. Citado 3 vezes nas páginas 45, 49 e 54.

YOURDON, E. *Modern Structured Analysis*. [S.l.]: Yourdon Press, 1989. Citado na página 26.

YSKOUT, K. et al. Threat modeling: From infancy to maturity. In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*. Seoul, South Korea: [s.n.], 2020. Citado 8 vezes nas páginas 18, 25, 27, 109, 111, 112, 113 e 114.

Apêndices

APÊNDICE A – TERMO DE CONSENTIMENTO DA PESQUISA

Título da Pesquisa: Threat Copilot: Um sistema de recomendação para modelagem de ameaças de software

Pesquisadores: Yuri Feitosa Negócio
Representante da Empresa: Organização X

Prezando(a) participante:

Sou aluno do Programa de Mestrado em Tecnologia da Informação do Instituto Federal da Paraíba e atualmente estou conduzindo uma pesquisa sob a orientação da Professora Doutora Juliana Dantas Ribeiro Viana de Medeiros. O objetivo da pesquisa é desenvolver um sistema de recomendação automatizado para a realização de análise de ameaças e a validação deste sistema por profissionais da área de segurança da informação.

Espera-se que a pesquisa proporcione uma nova abordagem sobre a automação da modelagem de ameaças em sistemas de software, facilitando o trabalho dos analistas de segurança da informação. A expectativa é que seja observado, além da facilidade de uso do sistema, a relevância dos resultados obtidos pelo seu uso.

A participação neste estudo é completamente opcional e voluntária, portanto, se em algum momento você decidir não participar ou desejar desistir da pesquisa, tem total liberdade para fazê-lo.

Todas as informações coletadas nesta pesquisa são altamente confidenciais. Apenas o pesquisador, a orientadora e a coorientadora terão acesso a esses dados. Na divulgação dos resultados, não haverá identificação individual dos participantes.

dos resultados, sua identidade e da empresa serão mantidas em sigilo absoluto, e todas as informações que possam identificá-lo serão omitidas.

Embora não haja benefícios diretos em participar desta pesquisa, você estará indiretamente contribuindo para o avanço do conhecimento científico e para uma melhor compreensão do fenômeno estudado, tanto no meio acadêmico quanto empresarial.

Sua participação nesta pesquisa não implicará em nenhum tipo de despesa ou pagamento.

Quaisquer dúvidas relativas à pesquisa poderão ser esclarecidas através dos contatos a seguir:

Pesquisador	
E-mail	negocio.yuri@academico.ifpb.edu.br
Telefone	+55 83 99838-7118

Orientadora	
E-mail	juliana.medeiros@ifpb.edu.br
Telefone	+55 83 98897-1168

Coorientadora	
E-mail	nadja.rodrigues@ifpb.edu.br
Telefone	+55 83 9309-9294

Após a apresentação dessas informações, pedimos sua permissão livre e voluntária para participar desta pesquisa, como segue:

> "Considerando as informações fornecidas acima, eu concordo voluntariamente em participar da pesquisa de forma esclarecida e consciente. Confirmo que recebi uma cópia deste termo de consentimento e autorizo a realização da pesquisa, bem como a divulgação dos resultados deste estudo nos moldes anteriormente descritos."

Assinatura do Representante da Empresa

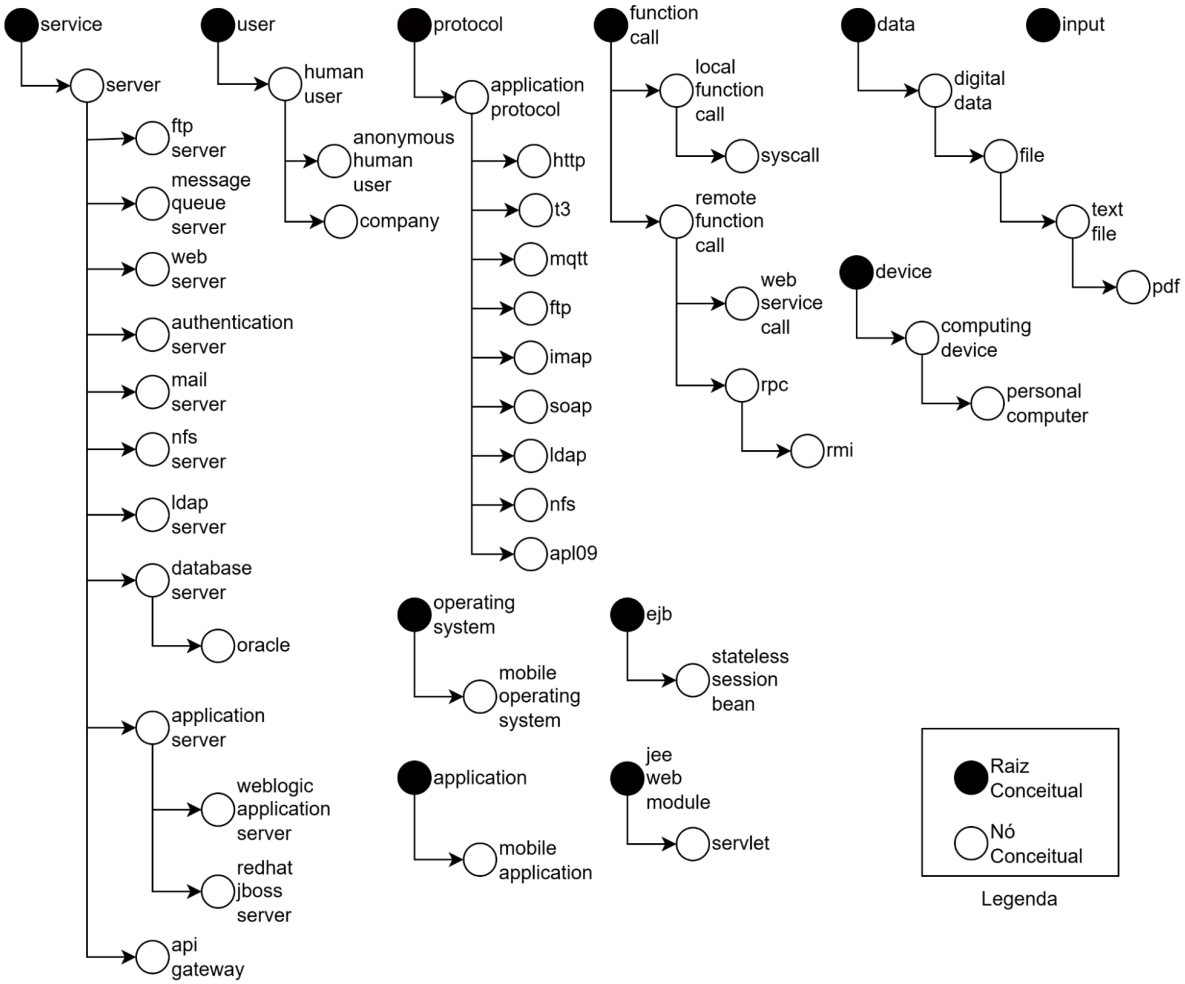
Assinatura da Orientadora

Assinatura da Corientadora

APÊNDICE B – VOCABULÁRIO (WORDNET)

Este apêndice apresenta na Figura 44 a WordNet inicial definida a partir dos termos rotulados nos modelos de ameaça para esta pesquisa.

Figura 44 – WordNet de domínio de Modelagem de Ameaças



Fonte: Autor

A Figura 44 apresenta o vocabulário utilizado como uma hierarquia de uma taxonomia. A raiz conceitual é representado como o um círculo preenchido de preto e representa a raiz conceitual do termo. O nó conceitual são as especializações dos termos.

APÊNDICE C – MODELO BASE DE CONHECIMENTO SOBRE CONTROLES DE SEGURANÇA

Lista de Controles de Segurança

- **DTPCS-A1**: Definição de requisitos de dimensionamento de cargas e picos adequados para a capacidade de atendimento do produto.
- **DTPCS-A2**: Redundância dos componentes do produto.
- **DTPCS-A3**: Elasticidade na capacidade da arquitetura do produto.
- **DTPCS-A4**: Implementar capacidade de degradação adequada (*graceful degradation*) do produto para cargas acima das estimadas.
- **DTPCS-A5**: Identificar e permitir apenas acesso autenticado aos recursos caros em termos de processamento, banda e armazenamento.
- **DTPCS-A6**: Proibir acesso anônimo aos recursos caros em termos de processamento, banda e armazenamento.
- **DTPCS-A7**: Verificar a disponibilidade do Web Application Firewall (WAF) e solicitar a configuração de limites e frequências adequadas para proteção contra DoS.
- **DTPCS-A8**: Identificar pontos do produto com consumo significativo de recursos e implementar controles ANTIBOT (rate-limit e CAPTCHA, quando aplicável).
- **DTPCS-A9**: Rotacionar logs de disco dos produtos.
- **DTPCS-B1**: Definir canais seguros criptografados (SSL/TLS) para comunicação entre elementos arquiteturais e atores do sistema.
- **DTPCS-C1**: Definir os componentes de autenticação e autorização do sistema.
- **DTPCS-C2**: Obrigar autenticação e identificação de todos os usuários, humanos ou sistemas, incluindo acessos anônimos.
- **DTPCS-C3**: Identificar usuários anônimos por endereços IP e mecanismos de identificação única.
- **DTPCS-C4**: Implementar autenticação multifator (MFA) para usuários privilegiados ou externos.
- **DTPCS-D1**: Definir o sistema de autorização de acesso do produto.
- **DTPCS-D2**: Aplicar o princípio do privilégio mínimo na definição de papéis e permissões.
- **DTPCS-D3**: Obrigar verificação de autorização por meio de políticas de controle de acesso.

- **DTPCS-D4**: Utilizar fontes confiáveis para obtenção de informações do usuário, como sessões no servidor ou JWT assinado.
- **DTPCS-E1**: Registrar logs de auditoria de negócio suficientes para identificação e compreensão das ações dos usuários.
- **DTPCS-E2**: Incluir identificadores nos logs para permitir correlação entre diferentes camadas e aplicações.
- **DTPCS-E3**: Identificar usuários nos logs por meio de identificadores controlados ou mascarados.
- **DTPCS-E4**: Definir requisitos de retenção de logs considerando a base legal.
- **DTPCS-E5**: Incluir informações básicas nos logs, como origem e horário.
- **DTPCS-E6**: Sincronizar os relógios de todos os componentes que produzem logs.
- **DTPCS-E7**: Integrar logs ao centralizador de logs.
- **DTPCS-E8**: Padronizar logs conforme a definição técnica ARATI.
- **DTPCS-F1**: Validar todos os dados de entrada em qualquer canal (formulários, URLs, APIs, parâmetros).
- **DTPCS-F2**: Validar dados de entrada de forma independente em cada camada do produto.
- **DTPCS-G1**: Não desabilitar controles de segurança em ambientes de desenvolvimento ou homologação.
- **DTPCS-G2**: Criar testes automatizados que impeçam a entrada em produção com controles desabilitados.
- **DTPCS-H1**: Verificar disponibilidade do WAF e solicitar inclusão do produto na proteção ativa.
- **DTPCS-I1**: Utilizar PKCE para prevenir vazamento de tokens no processo de autenticação.
- **DTPCS-J1**: Em eventos de monitoramento externos, incluir apenas dados de diagnóstico, excluindo informações identificáveis.
- **DTPCS-K1**: Não incluir informações sensíveis no empacotamento da aplicação (senhas, tokens ou chaves).
- **DTPCS-L1**: Usar assinatura digital para garantir a integridade de documentos exportados.
- **DTPCS-M1**: Controle em branco.
- **DTPCS-N1**: Verificar disponibilidade do WAF e solicitar inclusão na proteção ativa.
- **DTPCS-N2**: Implementar telas de erro adequadas.
- **DTPCS-N3**: Remover versões de produtos dos servidores de frontend.
- **DTPCS-O1**: Verificar *mime-type* de arquivos enviados por inspeção de conteúdo.

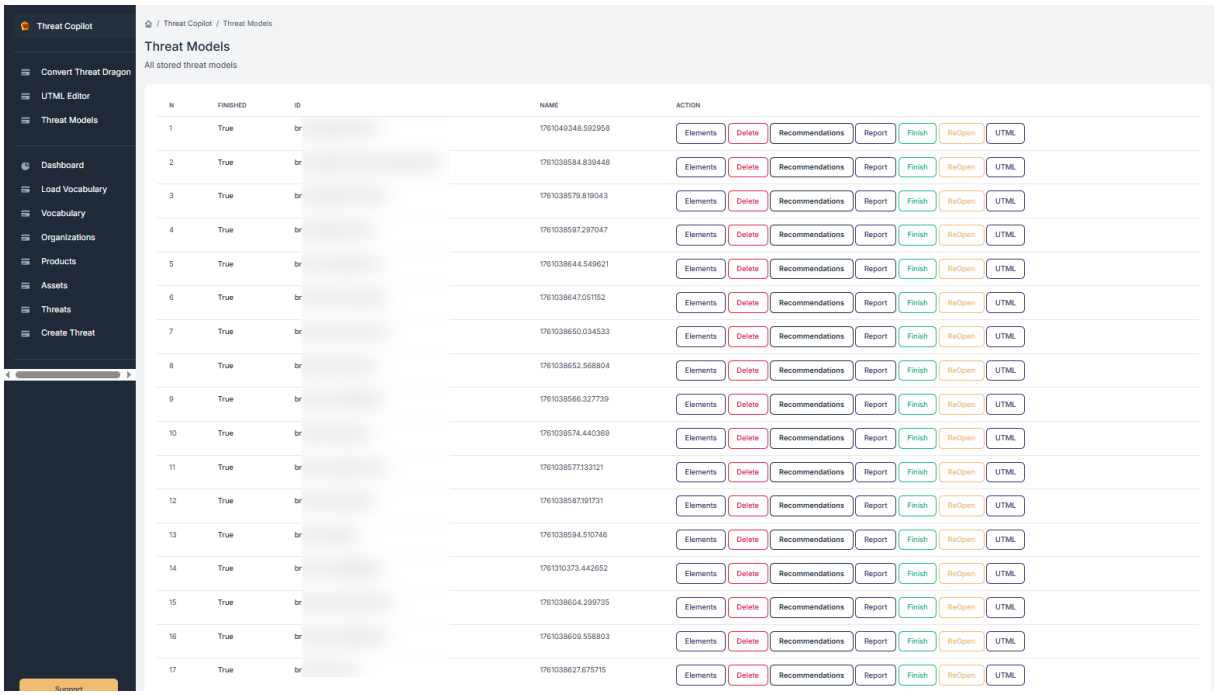
- **DTPCS-O2**: Implementar verificação de código malicioso nos dados enviados.
- **DTPCS-P1**: Automatizar publicação, agendamento e execução de jobs.
- **DTPCS-P2**: Assinar executáveis de jobs quando não houver automação.
- **DTPCS-Q1**: Proteger pontos que expõem dados pessoais contra scraping com controles ANTIBOT.
- **DTPCS-R1**: Implementar autenticação e autorização em serviços internos.
- **DTPCS-R2**: Implementar regras de firewall entre componentes.
- **DTPCS-R3**: Revisar acessos às interfaces administrativas.
- **DTPCS-S1**: Exigir certificados digitais válidos no servidor de autenticação.
- **DTPCS-S2**: Impedir redirecionamentos baseados em informações não confiáveis.
- **DTPCS-S3**: Monitorar serviços similares de autenticação (brand protection).
- **DTPCS-U1**: Implementar controle de autorização no backend para cada solicitação (proteção contra IDOR).
- **DTPCS-U2**: Aplicar menor privilégio e separação de funções.
- **DTPCS-U3**: Adotar modelo centralizado de autorização (RBAC, ABAC ou PBAC).
- **DTPCS-U4**: Utilizar identificadores opacos em requisições.
- **DTPCS-U5**: Restringir endpoints sensíveis via WAF e ACLs.
- **DTPCS-U6**: Validar contexto (ownership, tenant ou escopo) nas APIs.
- **DTPCS-U7**: Executar revisões de código e testes focados em autorização.
- **DTPCS-U8**: Aplicar MFA para operações críticas.
- **DTPCS-U9**: Registrar e monitorar tentativas de acesso negado.
- **DTPCS-V1**: Evitar armazenar senhas em locais acessíveis ou com permissões excessivas.
- **DTPCS-V2**: Armazenar senhas com algoritmos seguros (bcrypt, Argon2id, PBKDF2).
- **DTPCS-V3**: Executar testes específicos para armazenamento de credenciais.
- **DTPCS-V4**: Segregar segredos do código-fonte e configurações.
- **DTPCS-V5**: Criptografar segredos em repouso usando HSM, KMS ou cofres de segredos.
- **DTPCS-V6**: Evitar algoritmos criptográficos obsoletos.
- **DTPCS-W1**: Evitar expressões regulares de complexidade exponencial.
- **DTPCS-W2**: Utilizar regex compiladas e evitar construções ambíguas.

- **DTPCS-W3**: Realizar auditorias e testes de carga em regex.
- **DTPCS-W4**: Executar regex custosas em threads isoladas com timeout.
- **DTPCS-X1**: Definir periodicidade mínima de backups conforme RPO e RTO.
- **DTPCS-X2**: Executar testes periódicos de restauração.
- **DTPCS-Y1**: Configurar headers contra clickjacking.
- **DTPCS-Y2**: Utilizar certificados digitais válidos e confiáveis.
- **DTPCS-Y3**: Validar redirecionamentos para evitar Open Redirect.
- **DTPCS-Y4**: Detectar homógrafos e validar URLs.
- **DTPCS-Y5**: Aplicar HSTS.
- **DTPCS-Y6**: Monitorar domínios similares ou maliciosos.
- **DTPCS-Z1**: Configurar corretamente headers HTTP de segurança.
- **DTPCS-AA1**: Utilizar solução de segurança para comunicação B2B.
- **DTPCS-AB1**: Implementar autenticação mútua na comunicação entre produtos.
- **DTPCS-AC1**: Utilizar mecanismo padronizado de transmissão de arquivos via job.
- **DTPCS-AD1**: Restringir comunicação por microsegmentação de rede.
- **DTPCS-AE1**: Verificar integridade e origem de tokens JWT.
- **DTPCS-AF1**: Verificar autenticidade de tokens de credenciais.
- **DTPCS-AG1**: Evitar uso de dados pessoais como chaves primárias.
- **DTPCS-AH1**: Identificar usuários em logs por identificadores controlados ou mascarados.
- **DTPCS-AI1**: Utilizar Criptografia de Dados Transparente (TDE) no banco de dados.

APÊNDICE D – TELAS DA FERRAMENTA THREAT COPILOT

Este apêndice apresenta as telas principais da ferramenta Threat Copilot. Seu objetivo é auxiliar a compreensão do seu funcionamento por parte do leitor.

Figura 45 – Tela Principal da Ferramenta.



Fonte: Autor

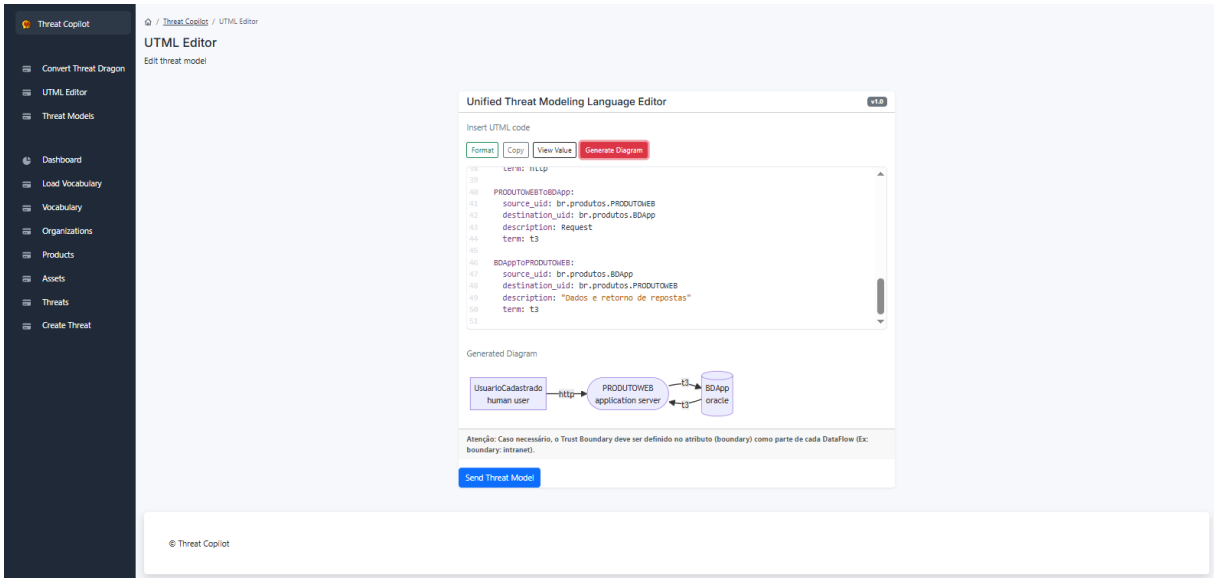
Na Figura 45 é apresentada a tela principal da ferramenta, ao lado esquerdo é o menu de ações na ferramenta e ao lado direito é a lista ordenada de modelos de ameaças existentes na base de conhecimento. Para cada linha, ou seja, para cada modelo são exibidos botões que representam as ações de visualização (Elements), remoção (Delete), execução das recomendações (Recommendations), emissão de relatórios (Report), finalização do modelo (Finish), reabertura do modelo (ReOpen) e emissão do YAML na linguagem UTML.

A Figura 46 apresenta a tela de edição do modelo de ameaças na linguagem UTML. Ao centro da tela é um editor de YAML que renderiza diretamente o modelo do diagrama de fluxo de dados em uma imagem abaixo do editor YAML. A sua função é permitir alterações antes de inserir o modelo de ameaças na base de dados de grafos.

A Figura 47 apresenta a tela de relatório. Ao lado esquerdo é apresentado um editor de texto na linguagem Markdown e ao lado direito é apresentado a versão renderizada em HTML do relatório de ameaças.

A Figura 48 apresenta a tela da lista de termos do vocabulário utilizado pela ferramenta Threat

Figura 46 – Tela de Edição de Modelos UTML.

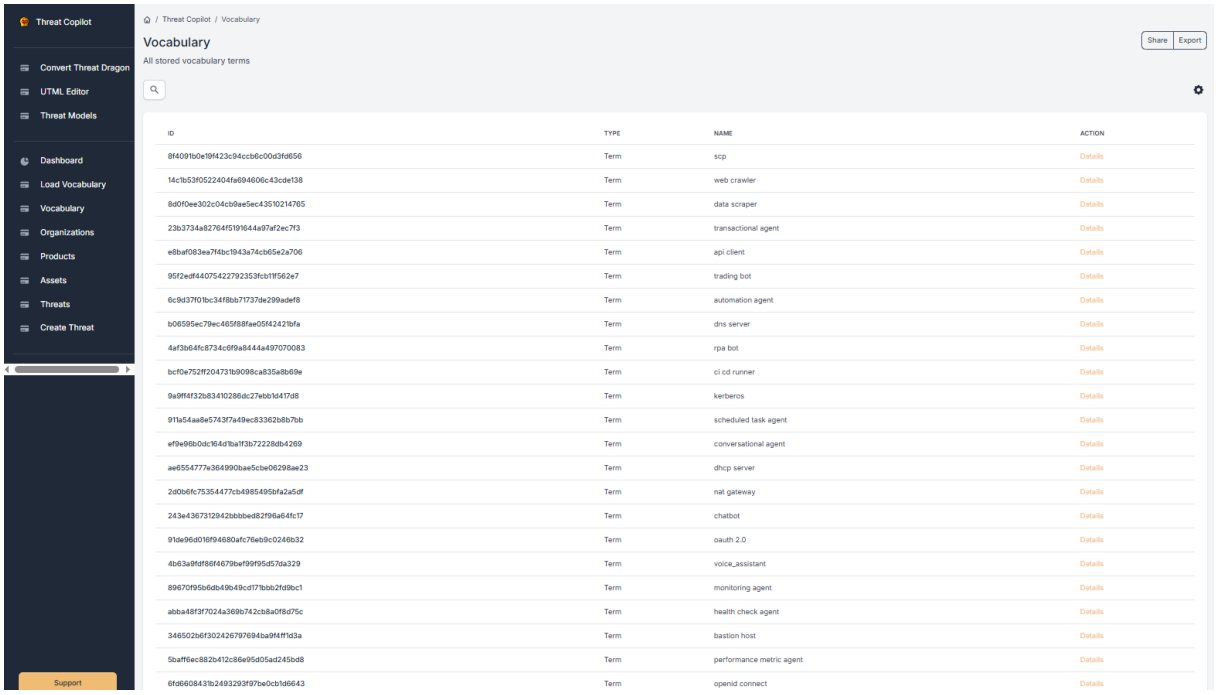


Fonte: Autor

Figura 47 – Tela de Edição de Relatório de Ameaças.

Fonte: Autor

Figura 48 – Tela de Visualização dos Vocabulários.



Fonte: Autor

Copilot. Ao centro é apresentado o vocabulário com a possibilidade detalhar a sua descrição através do link de Detalhes (Details).

APÊNDICE E – DESCRIÇÃO DO SISTEMA PARA O EXERCÍCIO

Este apêndice descreve o funcionamento do sistema utilizado como base para o exercício de modelagem de ameaças realizado pelos participantes da avaliação centrada no usuário, conduzida com base no método TAM–TTF.

Visão Geral

A aplicação de *Internet Banking* é uma plataforma online que permite aos clientes de um banco acessarem e gerenciarem suas contas e serviços financeiros de forma segura, prática e remota. O sistema é acessado via navegador web por clientes, funcionários e administradores da instituição.

Principais Funcionalidades

Consulta de Saldo e Extrato

Permite que o cliente visualize o saldo atual e o histórico de transações de suas contas, incluindo contas correntes, poupança, investimentos e cartões. As consultas podem ser exportadas em formato PDF.

Transferências

- Transferências entre contas do mesmo banco (transferência interna);
- Transferências entre bancos diferentes (TED, PIX, DOC, entre outros);
- Programação de transferências futuras ou recorrentes.

Gestão de Contas

- Alteração de dados cadastrais, como endereço, telefone e e-mail;
- Gerenciamento de cartões, incluindo ativação, bloqueio e solicitação de segunda via;
- Consulta e contratação de produtos e serviços, tais como seguros, empréstimos e investimentos;
- Configuração de notificações e definição de limites de transações.

Perfis de Usuário

Cliente

Usuário final da plataforma, com acesso restrito às suas próprias contas e produtos financeiros.

Atendente

Funcionário do banco com acesso a funcionalidades de suporte ao cliente, podendo visualizar contas, desbloquear acessos e auxiliar em operações, sempre com restrições de permissão e registro em logs de auditoria.

Administrador

Usuário com alto nível de privilégio, responsável pela gestão e configuração da plataforma, incluindo a criação de usuários, análise de logs, definição de permissões, manutenção e monitoramento do ambiente.

Componentes do Sistema

Aplicação Web (Frontend e Backend)

Frontend

Interface web responsiva fornecida por um servidor *Nginx*, acessível via navegador. O frontend é desenvolvido em *React* e é responsável pela interação com o usuário e pelo encaminhamento das requisições por meio de APIs.

Backend

Servidor de aplicação desenvolvido em *Java*, responsável por processar as requisições, validar regras de negócio, realizar a integração com o banco de dados e sistemas de terceiros, bem como gerenciar as sessões dos usuários.

Banco de Dados

Sistema gerenciador de banco de dados relacional *Oracle*, responsável por armazenar todas as informações críticas do sistema, incluindo:

- Dados de clientes;
- Transações bancárias;
- Dados cadastrais;
- Configurações e permissões.

APÊNDICE F – RESPOSTAS ÀS QUESTÕES ABERTAS DA AVALIAÇÃO TAM–TTF

Este apêndice apresenta as respostas às questões abertas feitas pelos participantes do exercício de avaliação centrada no usuário, conduzido com base no método TAM–TTF.

Pergunta 1

O que você achou da proposta da automação da modelagem de ameaças? (Considere em termos de facilidade, dificuldades e perspectiva de uso)

Respostas

- Muito importante, e que deve ser utilizada cada vez mais na Organização.
- É uma proposta válida e que agrega valor.
- Achei promissor.
- Parece ter um grande potencial para ajudar na difícil tarefa de modelar ameaças. Precisa ser mais testada.
- Ter uma base de conhecimento, com indicação de um *baseline* de ameaças, facilita muito a atividade. A parte de selecionar ou excluir as ameaças indicadas ficou de muito fácil uso.
- A parte de já ter a recomendação e controles também é essencial. Poder partir de um modelo gráfico de fluxo facilita o entendimento.
- A ferramenta é intuitiva e o processo para transformar o modelo DFD em JSON também. A facilidade de poder ter um banco de ameaças já mapeadas é interessante.
- Após a conversão, a visualização do modelo, o apontamento de ameaças e as recomendações facilitam a identificação de requisitos. Por ser uma ferramenta que torna ágil a modelagem, torna-se importante o seu uso por parte de especialistas em segurança.
- Muito bom, acredito que com um time experiente e havendo uma etapa para balizar a modelagem é possível acelerar as entregas.
- A proposta tem o intuito de padronizar e oferecer um norte para a modelagem de ameaças de forma prática e rápida, mas ainda necessita de conhecimentos prévios de um analista com experiência em análise de riscos.
- Ótimo conceito inicial. É possível imaginar várias formas de evolução da solução para torná-la ainda mais eficaz.

- A proposta agrega um valor incomensurável no desafiador dia a dia das equipes de segurança de uma empresa do nosso porte e responsabilidade. Sensacional.
- Prática, embora tenha o custo da modelagem inicial no padrão adequado.

Pergunta 2

Quais melhorias podem ser realizadas na proposta apresentada?

Respostas

- Reduzir a quantidade de *tags* possíveis, pois algumas parecem semanticamente muito próximas (por exemplo, *Human User* e *User*). Muitas opções podem gerar confusão para usuários iniciantes.
- Verificar uma forma de que modelos de ameaças antigos não sejam considerados ou recebam um peso menor na análise.
- Possibilitar a realimentação das vulnerabilidades já tratadas, de modo que não apareçam novamente no relatório.
- Gerar estatísticas a partir das ameaças identificadas (mais recorrentes, mais recorrentes por tipo de aplicação etc.).
- Algumas ameaças catalogadas parecem ser muito genéricas.
- Acrescentar o diagrama ao relatório final.
- Incluir categoria na listagem de ameaças para facilitar a busca. Uma padronização textual das ameaças também é importante para evitar erros de interpretação.
- Melhorar a navegação, permitindo acessar uma "página do modelo" para navegar entre elementos, relatórios e recomendações sem retornar à página principal.
- Integração com outras ferramentas de modelagem.
- Apresentar requisitos de segurança após a identificação das ameaças, tornando a modelagem completa em seu ciclo de segurança.
- Melhorar a parte das *tags*, preferencialmente com uma funcionalidade no Threat Dragon que elimine a necessidade de consultas externas.
- Apresentar métodos manuais de modelagem de ameaças e compará-los com a eficiência do método automatizado.
- O Threat Dragon apresenta alguns problemas de usabilidade. Um editor de diagramas STRIDE mais maduro, integrado ao Threat Copilot, seria uma evolução significativa.
- Disponibilizar a interface no idioma Português do Brasil.
- Automatizar a geração das *tags* com base no contexto.