



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO SISTEMAS DE INFORMAÇÃO

Julia Knabem

**Guia de Diretrizes de Segurança para o Desenvolvimento de Aplicações
Governamentais**

Florianópolis
2026

Julia Knabem

**Guia de Diretrizes de Segurança para o Desenvolvimento de Aplicações
Governamentais**

Trabalho de Conclusão de Curso submetido ao curso de Sistemas de Informação do Centro Campus de Florianópolis da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof^a. Thaís Bardini Idalino, Dra.

Florianópolis
2026

Ficha catalográfica para trabalhos acadêmicos

Julia Knabem

**Guia de Diretrizes de Segurança para o Desenvolvimento de Aplicações
Governamentais**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de
Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de em
Sistemas de Informação

Florianópolis, _____ de _____ de 2026.

Coordenação do Curso

Banca examinadora

Profª. Thaís Bardini Idalino, Dr.(a)
Orientador(a)

Prof.(a) Jean Carlo Rossa Hauck, Dr.(a)
Instituição UFSC

Prof.(a) Jean Everson Martina, Dr.(a)
Instituição UFSC

Florianópolis, 2026.

Dedico este trabalho, e todas as minhas conquistas acadêmicas, ao meu pai, Rogério Knabem, por me ensinar que o conhecimento é o bem mais precioso que eu poderia ter.

AGRADECIMENTOS

Ao meu companheiro, Luiz Alfredo, que ressignificou para mim a palavra 'companhia', mostrando que estar junto também é ação, apoio e cuidado, e que foi fundamental para que este trabalho se tornasse possível. Ao meu pai, Rogério, por me incentivar nos estudos desde cedo. À minha mãe, Rozeli, por me ensinar a nunca depender de outras pessoas. À minha irmã Bruna, por abrir caminhos e por ser a minha primeira referência de que eu poderia chegar onde quisesse. À minha orientadora, por ir além em uma época que nem o mínimo costuma ser feito. Ao Paulo que, apesar de não estar mais aqui, continua sendo um dos maiores exemplos de dedicação que já conheci. E àqueles que me possibilitaram e me incentivaram a chegar até aqui. A todos, meu muito obrigada.

RESUMO

Este trabalho tem como objetivo o desenvolvimento de um guia prático de diretrizes de segurança da informação voltado a desenvolvedores de aplicações governamentais, estruturado segundo as fases do ciclo de vida de desenvolvimento de *software* (SDLC). A motivação parte da crescente digitalização dos serviços públicos brasileiros, do avanço da Lei Geral de Proteção de Dados Pessoais (LGPD) e da dificuldade prática de traduzir normas e *frameworks* em ações concretas no cotidiano do desenvolvimento. A metodologia adotada consiste em revisão da literatura que combina documentos regulatórios nacionais (como a LGPD, o Guia de Requisitos Mínimos do Governo Federal e as Boas Práticas em Segurança da Informação do TCU), *frameworks* internacionais amplamente reconhecidos (ISO/IEC 27001, NIST *Cybersecurity Framework*, OWASP *Secure by Design* e OWASP *Secure Coding Practices*), e publicações acadêmicas relacionadas ao desenvolvimento seguro de *software*. A partir de uma análise comparativa dessas fontes, foram identificadas e priorizadas diretrizes aplicáveis às fases de especificação de requisitos, projeto, codificação, testes e implantação e manutenção, com base em critérios de impacto na segurança, aplicabilidade ao SDLC, relevância normativa, recorrência entre as fontes e adequação ao contexto governamental. O guia resultante foi disponibilizado como artefato independente em repositório aberto, com interface *web* que permite consulta direta por fase do desenvolvimento, sem exigir a leitura integral deste trabalho. A avaliação foi conduzida por meio de mapeamento sistemático com os requisitos de Nível 1 do OWASP *Application Security Verification Standard* (ASVS) 5.0.0, instrumento inédito no processo de construção do guia, garantindo independência avaliativa. Dos 65 requisitos analisados, 36 foram classificados como cobertos (55,4%), 16 como parcialmente cobertos (24,6%) e 13 como não cobertos (20%), com cobertura de 100% nos temas de “Criptografia”, “Comunicação segura”, “Configuração” e “Codificação e Arquitetura Seguras”. Os resultados indicam que o guia atende de forma satisfatória às categorias centrais de segurança no desenvolvimento e aponta lacunas específicas (como tratamento de arquivos e serviços *web*) que representam oportunidades para trabalhos futuros. Assim, o trabalho contribui para o fortalecimento da cultura de segurança nos serviços digitais públicos, reduzindo riscos associados ao tratamento de dados pessoais e ampliando a capacidade das equipes de desenvolvimento de atender às demandas regulatórias, técnicas e operacionais do Governo Federal.

Palavras-chave: segurança da informação; desenvolvimento seguro; LGPD; proteção de dados pessoais; SDLC; aplicações governamentais; OWASP; segurança por design.

ABSTRACT

This work aims to develop a practical security guideline directed at developers of Brazilian Government applications, structured according to the phases of the Software Development Life Cycle (SDLC). The motivation stems from the growing digitalization of public services in Brazil, the advancement of the General Data Protection Law (LGPD), and the practical difficulty of translating established standards and frameworks into concrete actions in the day-to-day development workflow. The methodology consists of a literature review combining national regulatory documents (including the LGPD, the Federal Government's Minimum Requirements Guide, and the TCU Information Security Best Practices), internationally recognized frameworks (ISO/IEC 27001, the NIST Cybersecurity Framework, OWASP Secure by Design, and OWASP Secure Coding Practices), and academic publications related to secure software development. Through a comparative analysis of these sources, guidelines applicable to the phases of requirements specification, design, coding, testing, and deployment and maintenance were identified and prioritized based on criteria of security impact, SDLC applicability, normative relevance, recurrence across sources, and suitability for the governmental context. The resulting guideline was made available as an independent artifact in an open repository, with a web interface enabling direct consultation by development phase, without requiring a full reading of this work. Validation was conducted through a systematic mapping against the Level 1 requirements of the OWASP Application Security Verification Standard (ASVS) 5.0.0, an instrument not used during the guideline's construction, ensuring evaluative independence. Of the 65 requirements analyzed, 36 were classified as covered (55.4%), 16 as partially covered (24.6%), and 13 as not covered (20%), with 100% full coverage achieved in "Cryptography", "Secure Communication", "Configuration", and "Secure Coding and Architecture". The results indicate that the guideline satisfactorily addresses the core security categories in software development and highlights specific gaps (such as file handling and API security) that represent opportunities for future work. Thus, this work contributes to strengthening the security culture in public digital services, reducing risks associated with personal data processing and enhancing the capacity of development teams to meet the regulatory, technical, and operational demands of the Federal Government.

Keywords: information security; secure development; LGPD; personal data protection; SDLC; government applications; OWASP; secure by design.

LISTA DE FIGURAS

Figura 1 – Comparativo entre o OWASP Top 10 de 2021 e 2025.....	42
Figura 2 – Tela inicial do guia.....	80
Figura 3 – Tela de listagem de diretrizes de uma fase, segundo nível de navegação	81
Figura 4 – Tela de detalhe de uma diretriz, terceiro nível de navegação	82
Figura 5 – Tela de detalhe de uma diretriz, terceiro nível de navegação	83
Figura 6 – Tela de fontes utilizadas.....	84

LISTA DE TABELAS

Tabela 1 – Ranking de capítulos do ASVS por cobertura do guia.....	86
--	----

LISTA DE QUADROS

Quadro 1 – Critérios de priorização de diretrizes	25
Quadro 2 – Matriz comparativa: Gestão de identidade e controle de acesso	51
Quadro 3 – Matriz comparativa: Proteção de dados e privacidade	53
Quadro 4 – Matriz comparativa: Validação de entradas e segurança na implementação	56
Quadro 5 – Matriz comparativa: Gestão de vulnerabilidades e atualizações	58
Quadro 6 – Matriz comparativa: Arquitetura e design seguro	60
Quadro 7 – Matriz de priorização de diretrizes	66
Quadro 8 – Matriz de cobertura frente ao OWASP Top 10	75

LISTA DE ABREVIATURAS E SIGLAS

ANPD	Autoridade Nacional de Proteção de Dados
API	<i>Application Programming Interface</i> (Interface de programação de aplicações)
ASVS	<i>Application Security Verification Standard</i> (Padrão de verificação de segurança de aplicativos)
CSF	<i>Cybersecurity Framework</i> (Framework de cibersegurança)
CVE	<i>Common Vulnerabilities and Exposures</i> (Vulnerabilidades e exposições comuns)
DDoS	<i>Distributed Denial of Service</i> (Negação de Serviço Distribuído)
GSI	Gabinete de Segurança Institucional
GSI/PR	Gabinete de Segurança Institucional da Presidência da República
IBM	<i>International Business Machines Corporation</i>
LGPD	Lei Geral de Proteção de Dados Pessoais
NIST	<i>National Institute of Standards and Technology</i> (Instituto Nacional de Padrões e Tecnologia)
OWASP	<i>Open Web Application Security Project</i> (Projeto de Segurança de Aplicações Web Abertas)
PCN	Plano de Continuidade do Negócio
PPSI	Programa de Privacidade e Segurança da Informação
RBAC	<i>Role-Based Access Control</i> (Controle de Acesso baseado em Roles/Papéis)
SCA	<i>Software Composition Analysis</i> (Análise de composição de <i>software</i>)
SDLC	<i>Software Development Life Cycle</i> (Ciclo de vida de desenvolvimento de <i>software</i>)
S-SDLC	<i>Secure Software Development Life Cycle</i> (Ciclo de vida de desenvolvimento de <i>software</i> seguro)
SQL	<i>Structured Query Language</i> (Linguagem de consulta estruturada)
TCC	Trabalho de Conclusão de Curso
TCU	Tribunal de Contas da União
TLS	<i>Transport Layer Security</i> (Segurança da camada de transporte)
UFSC	Universidade Federal de Santa Catarina
VPN	<i>Virtual Private Network</i> (Rede Privada Virtual)

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	JUSTIFICATIVA E CONTEXTUALIZAÇÃO DA PROPOSTA	17
1.2	OBJETIVOS	18
1.2.1	Objetivos Gerais.....	18
1.2.2	Objetivos Específicos	18
2	METODOLOGIA	20
2.1	SELEÇÃO DAS FONTES	20
2.2	MÉTODO DE ANÁLISE E COMPARAÇÃO DE DIRETRIZES	21
2.3	CRITÉRIOS DE PRIORIZAÇÃO	24
2.3.1	Impacto na segurança	25
2.3.2	Aplicabilidade prática e aderência ao SDLC.....	25
2.3.3	Relevância normativa e recorrência nas fontes	26
2.3.4	Adequação ao contexto de aplicações governamentais	26
2.4	VERIFICAÇÃO DA COBERTURA FRENTE AO OWASP TOP 10	27
2.5	AVALIAÇÃO DO GUIA.....	27
3	FUNDAMENTAÇÃO TEÓRICA.....	29
3.1	FRAMEWORK	29
3.2	SEGURANÇA DE REDES, CIBERSEGURANÇA E SEGURANÇA DA INFORMAÇÃO	30
3.3	CONFIDENCIALIDADE, INTEGRIDADE E DISPONIBILIDADE	31
3.3.1	Confidencialidade	31
3.3.2	Integridade.....	32
3.3.3	Disponibilidade	32
3.4	DESENVOLVIMENTO DE SOFTWARE	33
3.4.1	Ciclo de vida de desenvolvimento de <i>software</i>.....	33
3.4.2	Ciclo de vida de desenvolvimento de <i>software</i> seguro.....	35
3.4.3	Segurança por <i>design</i>	36
3.4.4	Requisitos funcionais e não funcionais.....	36
3.5	DADOS E A LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS	37
3.5.1	Dados Pessoais.....	38
3.5.2	Dados Pessoais Sensíveis	38
3.5.3	Dados Públicos	39

3.6	TRANSPORT LAYER SECURITY (TLS)	39
3.7	MODELAGEM DE AMEÇAS (THREAT MODELING)	40
3.8	OWASP TOP 10	41
4	REVISÃO DA LITERATURA	43
4.1	METODOLOGIA DE SELEÇÃO DAS LITERATURAS	43
4.1.1	Identificação de documentos normativos e regulatórios nacionais	43
4.1.2	Levantamento de boas práticas e frameworks de segurança utilizados no setor privado e em organismos internacionais	44
4.1.3	Trabalhos acadêmicos relacionados	44
4.2	DOCUMENTOS NORMATIVOS E REGULATÓRIOS NACIONAIS	44
4.2.1	Lei geral de proteção de dados pessoais (Lei nº 13.709/2018)	44
4.2.2	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (2023)	45
4.2.3	Boas Práticas em Segurança da Informação (TCU, 2012)	45
4.2.4	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	46
4.3	FRAMEWORKS E BOAS PRÁTICAS DO SETOR PRIVADO E INTERNACIONAL	46
4.3.1	OWASP Secure by Design Framework (2025)	46
4.3.2	ISO/IEC 27001 (2023)	47
4.3.3	NIST Cybersecurity Framework (2024)	47
4.4	TRABALHOS ACADÊMICOS RELACIONADOS	47
4.4.1	Arquitetura integrada de normas e frameworks de desenvolvimento de software seguro aplicada para apoiar a identificação de falhas na especificação de requisitos (GATTO, 2024)	48
4.5	INSTRUMENTO DE VERIFICAÇÃO	48
5	ANÁLISE E DISCUSSÃO	50
5.1	GESTÃO DE IDENTIDADE E CONTROLE DE ACESSO	51
5.2	PROTEÇÃO DE DADOS E PRIVACIDADE	53
5.3	VALIDAÇÃO DE ENTRADAS E SEGURANÇA NA IMPLEMENTAÇÃO	55
5.4	GESTÃO DE VULNERABILIDADES E ATUALIZAÇÕES	58
5.5	ARQUITETURA E DESIGN SEGURO	59
5.6	SÍNTESE DA ANÁLISE COMPARATIVA	62
6	DESENVOLVIMENTO DO GUIA	63

6.1	DECISÕES DE DESENVOLVIMENTO	63
6.2	APLICAÇÃO DOS CRITÉRIOS DE PRIORIZAÇÃO	64
6.3	BASE DE REFERÊNCIA DAS DIRETRIZES	73
6.4	VERIFICAÇÃO DE COBERTURA FRENTE AO OWASP TOP 10	74
6.5	FORMATO E DISPONIBILIZAÇÃO DO GUIA	78
6.6	INTERFACE E FLUXO DE NAVEGAÇÃO DO GUIA	79
7	AVALIAÇÃO DO GUIA	85
7.1	DISCUSSÃO DOS RESULTADOS	86
7.1.1	Áreas com cobertura satisfatória	86
7.1.2	Áreas parcialmente cobertas	87
7.1.3	Áreas não cobertas	88
7.1.4	Diretrizes sem citação ao longo da avaliação	88
7.1.5	Considerações gerais	89
8	CONCLUSÃO	90
	REFERÊNCIAS BIBLIOGRÁFICAS	92
	APÊNDICE A - ARTIGO BASEADO NESTE TRABALHO	95
	APÊNDICE B - MATRIZ COMPARATIVA DE DIRETRIZES	100
	APÊNDICE C - MATRIZ DE CORRESPONDÊNCIA COM O ASVS	101

1 INTRODUÇÃO

A crescente dependência de sistemas digitais para a prestação de serviços tornou a segurança da informação um elemento indispensável do desenvolvimento de *software*. Além disso, a Lei Geral de Proteção de Dados Pessoais (Lei nº 13.709/2018) (BRASIL, 2018) trouxe ao Brasil um marco regulatório essencial para a proteção da privacidade e do tratamento de dados. O artigo 52 da lei prevê que instituições, sejam públicas ou privadas, que não implementarem mecanismos adequados de proteção de dados estão sujeitas a riscos jurídicos, financeiros e reputacionais.

Nesse cenário, a segurança da informação deixa de ser um aspecto secundário para se tornar um requisito indispensável de qualquer aplicação. Questões como segurança de redes (garantia do tráfego seguro de informações), segurança da informação (armazenamento e manipulação correta dos dados) e segurança cibernética (defesa contra ataques e ameaças externas) precisam estar integradas ao ciclo de desenvolvimento de *software*, de forma a assegurar conformidade legal, integridade operacional e confiança dos usuários.

Para o setor público, esse desafio é ainda mais relevante. Aplicações governamentais¹ lidam frequentemente com dados pessoais sensíveis de milhões de cidadãos e, portanto, carregam uma grande responsabilidade. A ausência de diretrizes claras e práticas para equipes de desenvolvimento pode resultar em vulnerabilidades críticas, incidentes de segurança e, conseqüentemente, em perda de credibilidade.

Embora normas como a LGPD (BRASIL, 2018), a ISO/IEC 27001 (ABNT, 2022), o NIST *Cybersecurity Framework* (NIST, 2024) e o OWASP (OWASP, 2025b) sejam amplamente reconhecidas, a simples existência dessas referências não garante sua adoção no dia a dia do desenvolvimento. Pesquisas na área indicam que a principal barreira não é o desconhecimento das normas, mas a dificuldade de traduzi-las em ações concretas dentro das etapas do SDLC (GATTO, 2024). Um desenvolvedor que consulta a ISO/IEC 27001, por exemplo, encontrará controles organizacionais e de gestão, mas não encontrará orientação sobre como validar

¹ Para fins deste trabalho, entende-se por aplicação governamental qualquer sistema ou aplicação digital desenvolvida por ou para órgãos e entidades da Administração Pública nas esferas federal, estadual ou municipal. O termo abrange aplicações web, mobile e híbridas, independentemente da plataforma ou tecnologia adotada.

entradas de dados em uma API ou como estruturar permissões em um sistema de autenticação.

Diante desse cenário, este trabalho propõe a elaboração de um guia prático de requisitos de segurança para aplicações governamentais, a partir de um projeto em desenvolvimento pela Universidade Federal de Santa Catarina (UFSC), em parceria com o Ministério do Trabalho e Emprego (MTE), chamado Ecosol Digital. O guia visa consolidar e estruturar diretrizes de segurança da informação provenientes de diferentes normas, legislações e *frameworks* de forma prática, facilitando a integração e aplicabilidade da segurança da informação como elemento intrínseco ao desenvolvimento, oferecendo orientações acessíveis a desenvolvedores de diferentes níveis de maturidade e contribuindo para o fortalecimento da cultura de segurança nos serviços digitais do setor público.

1.1 JUSTIFICATIVA E CONTEXTUALIZAÇÃO DA PROPOSTA

A crescente digitalização dos serviços públicos no Brasil amplia significativamente a superfície de ataque a que sistemas governamentais estão expostos. Incidentes recentes evidenciam que falhas de segurança no desenvolvimento de *software* representam um risco real e de larga escala: no caso das fraudes no sistema do INSS, auditorias do Tribunal de Contas da União (TCU) identificaram, entre 2023 e 2024, o comprometimento de aproximadamente 400 credenciais de acesso e a presença de 60 dispositivos não autorizados na rede da Dataprev (TCU, 2024).

As investigações revelaram que o sistema processava descontos diretamente nos benefícios de aposentados e pensionistas sem qualquer verificação da autenticidade das autorizações, uma falha de controle de acesso que, se tratada ainda na fase de projeto do sistema, poderia ter evitado prejuízos estimados em R\$ 1,4 bilhão. Casos como este ilustram como a ausência de requisitos de segurança bem definidos desde as etapas iniciais do ciclo de vida de desenvolvimento de *software* pode resultar em impactos de larga escala sobre dados pessoais sensíveis de milhões de cidadãos.

Embora existam normas e *frameworks* consolidados, como a LGPD, a ISO/IEC 27001, o NIST *Cybersecurity Framework* e as diretrizes do OWASP, a aplicação prática dessas referências por equipes de desenvolvimento não é intuitiva.

Esses documentos, em sua maioria, possuem caráter normativo ou gerencial, sendo voltados a gestores e auditores, e não oferecem orientações diretas e aplicáveis ao cotidiano do desenvolvedor. Essa lacuna entre o que é regulamentado e o que é efetivamente implementado nas etapas do SDLC reforça a necessidade de um guia consolidado, acessível e orientado à prática.

No contexto do projeto Ecosol Digital (<https://ecosol.ufsc.br>), essa necessidade se torna ainda mais evidente: trata-se de uma aplicação governamental que tratará dados pessoais de trabalhadores da economia solidária, exigindo aderência às diretrizes do Governo Federal e conformidade com a LGPD desde as etapas iniciais de desenvolvimento. Esse cenário serviu como ponto de partida e referência contextual para este trabalho, mas não delimita seu escopo. A metodologia e as diretrizes propostas foram elaboradas de forma genérica e independente de arquiteturas ou decisões específicas daquele projeto, para que o guia resultante possa ser aplicado a qualquer iniciativa de desenvolvimento de aplicações no âmbito nacional, independentemente do estágio de maturidade da equipe ou das tecnologias adotadas.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

O objetivo deste trabalho é desenvolver um guia prático de requisitos e boas práticas de segurança da informação, direcionado a desenvolvedores de aplicações que desejam ou necessitam atender às diretrizes e regulamentações estabelecidas e/ou sugeridas pelo Governo Federal. O guia será estruturado de forma a integrar a segurança como parte intrínseca do ciclo de vida de desenvolvimento de software, e não como um elemento adicional, promovendo praticidade, aplicabilidade e conformidade com as demandas do setor público.

1.2.2 Objetivos Específicos

1. Mapear e analisar normas, legislações e boas práticas de segurança da informação aplicáveis ao desenvolvimento de aplicações no setor público,

incluindo LGPD, diretrizes do GSI, recomendações do TCU e o Guia de Requisitos Mínimos do Governo Federal.

2. Levantar requisitos de segurança essenciais para o desenvolvimento de aplicações governamentais, considerando riscos específicos de sistemas públicos.
3. Estruturar um fluxo de aplicação dos requisitos de segurança ao longo das etapas do ciclo de vida de desenvolvimento de *software* (SDLC), de forma a orientar a incorporação contínua da segurança desde a concepção até a implantação.
4. Propor um guia acessível e orientado a desenvolvedores, em formato simplificado, garantindo aplicabilidade mesmo em equipes com maturidade limitada em segurança.
5. Contribuir para a disseminação da cultura de segurança no desenvolvimento de soluções digitais, destacando a importância de integrar requisitos de segurança desde a fase inicial de projeto.

2 METODOLOGIA

Este trabalho adota uma abordagem metodológica de natureza qualitativa, com caráter exploratório e descritivo. A pesquisa é classificada como aplicada, uma vez que tem como objetivo a elaboração de um artefato prático (um guia de diretrizes de segurança da informação) com potencial de utilização por equipes de desenvolvimento.

A condução do estudo ocorre por meio de uma sequência estruturada de etapas, cada uma composta por atividades específicas de análise, consolidação e sistematização das informações. Essas etapas são acompanhadas pela elaboração contínua dos textos, permitindo o registro organizado das decisões metodológicas, bem como a análise dos resultados obtidos ao longo do processo.

Ainda que o Ecosol Digital tenha funcionado como ponto de partida e referência contextual, as diretrizes propostas neste trabalho foram elaboradas de forma genérica, sem vinculação a arquiteturas ou decisões específicas daquele projeto. O objetivo foi garantir que o guia resultante pudesse ser aplicado a qualquer projeto de desenvolvimento de aplicações no âmbito nacional, independentemente de seu estágio de maturidade ou das ferramentas adotadas pela equipe. Essa escolha metodológica amplia o alcance da proposta e evita que o trabalho se limite a um estudo de caso particular.

A seguir, são apresentadas as etapas e atividades que compõem a metodologia adotada nesta pesquisa.

2.1 SELEÇÃO DAS FONTES

A seleção das normas, legislações e *frameworks* considerados neste estudo foi realizada com base nos seguintes critérios:

- **Relevância normativa:** documentos oficialmente adotados ou recomendados no contexto do Governo Federal (ex.: LGPD, PPSI).
- **Adoção internacional:** *frameworks* amplamente utilizados e reconhecidos globalmente (ex.: NIST, OWASP).
- **Aderência ao desenvolvimento de *software*:** presença de diretrizes aplicáveis ao ciclo de vida de desenvolvimento (SDLC).

- **Atualidade:** utilização das versões mais recentes disponíveis até o momento da pesquisa.
- **Fundamentação acadêmica:** inclusão de trabalhos científicos relevantes que abordam o desenvolvimento seguro de *software*, a integração de normas e *frameworks* ou a aplicação de práticas de segurança ao longo do SDLC.

A seleção dessas fontes buscou equilibrar referências normativas, técnicas e acadêmicas, permitindo uma análise abrangente e fundamentada das diretrizes de segurança da informação.

Os documentos e trabalhos selecionados são descritos e analisados de forma mais detalhada na seção de Revisão da Literatura, onde são exploradas suas contribuições específicas para o desenvolvimento deste trabalho.

2.2 MÉTODO DE ANÁLISE E COMPARAÇÃO DE DIRETRIZES

Para análise das referências selecionadas, foi adotada uma abordagem comparativa baseada na identificação e categorização das diretrizes de segurança presentes em cada fonte.

As diretrizes extraídas das fontes selecionadas foram classificadas com base em aspectos técnicos e funcionais da segurança da informação no contexto do desenvolvimento de *software*. Essa categorização tem como objetivo permitir uma comparação estruturada entre as diferentes abordagens, reduzindo ambiguidades e sobreposições, conforme recomendado por abordagens de engenharia de *software* e análise de requisitos (SOMMERVILLE, 2019).

A definição das cinco categorias adotadas neste trabalho partiu de um critério estrutural, não quantitativo: foram selecionadas os documentos que organizam suas recomendações em agrupamentos funcionais explícitos reconhecidos internacionalmente. A ISO/IEC 27001 (ABNT, 2022) estrutura seus controles em 4 domínios temáticos. O NIST *Cybersecurity Framework* (NIST, 2024) organiza suas recomendações em funções (Identificar, Proteger, Detectar, Responder e Recuperar). E o OWASP *Secure by Design Framework* (OWASP, 2025b) agrupa seus princípios em domínios de *design* seguro. Os demais documentos citados na Revisão da Literatura deste trabalho apresentam recomendações relevantes, mas sem uma taxonomia própria de categorias que pudesse ser usada como referência de

agrupamento. A sobreposição entre os agrupamentos funcionais destes três documentos resultou em cinco áreas de convergência consistente, que foram adotadas como categorias de análise.

As categorias adotadas foram:

- **Gestão de identidade e controle de acesso:** engloba diretrizes relacionadas à autenticação, autorização, gestão de identidades e aplicação do princípio do menor privilégio, aspectos fundamentais para a proteção de sistemas e dados;
- **Proteção de dados e privacidade:** compreende diretrizes voltadas à coleta, armazenamento, processamento e compartilhamento de dados, incluindo criptografia, anonimização e minimização de dados, alinhadas aos princípios da LGPD (BRASIL, 2018) e ao conceito de *privacy by design* (PPSI, 2023);
- **Validação de entradas e segurança na implementação:** abrange práticas relacionadas à codificação segura, validação de entradas e mitigação de vulnerabilidades comuns, como injeção e *cross-site scripting*, conforme amplamente citado na tese de Gatto (GATTO, 2024);
- **Gestão de vulnerabilidades e atualizações:** inclui diretrizes voltadas à identificação, análise e correção de vulnerabilidades, bem como à gestão de atualizações e dependências, alinhadas às práticas de gestão contínua de riscos e vulnerabilidades descritas pelo NIST (2024) e pela ISO/IEC 27001 (ABNT, 2022);
- **Arquitetura e design seguro:** contempla princípios relacionados à modelagem de ameaças, definição de arquiteturas seguras e redução da superfície de ataque, conforme o conceito de *Security by Design* (DEOGUN et al., 2019; OWASP, 2025b);

Cada diretriz identificada foi alocada na categoria que melhor representa seu objetivo principal, sendo adotado o critério de predominância funcional nos casos em que uma diretriz poderia se enquadrar em mais de uma categoria. Essa abordagem é consistente com práticas de classificação em engenharia de *software*, nas quais requisitos são organizados conforme sua função predominante no sistema (SOMMERVILLE, 2019).

A seleção dos aspectos analisados dentro de cada categoria seguiu o critério de recorrência entre as fontes: foram incluídos apenas os aspectos que aparecem, de forma explícita ou implícita, em pelo menos dois dos documentos selecionados para

análise. Esse critério garante que a comparação seja tecnicamente significativa, uma vez que aspectos presentes em apenas uma fonte não permitem contraste entre abordagens, além de assegurar que os aspectos selecionados representem preocupações consolidadas na literatura de segurança, e não particularidades isoladas de uma única referência.

Para a classificação do nível de cobertura de cada fonte em relação a cada aspecto analisado foram adotados três níveis, definidos com base em dois critérios combinados: a intencionalidade do tratamento, isto é, se o documento aborda o aspecto como objetivo direto ou apenas como consequência de outro tema; e a especificidade da orientação oferecida, ou seja, se o documento fornece controles, requisitos ou recomendações aplicáveis de forma direta ou apenas princípios gerais.

A partir da combinação desses dois critérios, os níveis foram definidos da seguinte forma:

- **Aborda explicitamente:** o documento dedica controle, seção ou requisito nominalmente voltado ao aspecto analisado, com orientação técnica ou normativa suficiente para guiar sua implementação de forma direta;
- **Aborda parcialmente:** o documento menciona ou implica o aspecto dentro de um contexto mais amplo, sem dedicar controle ou seção específica a ele, ou trata apenas um subconjunto do aspecto analisado. Por exemplo, ao abordar a necessidade de controle de acesso baseado em papéis como resultado esperado, sem especificar o mecanismo de implementação;
- **Não aborda:** o documento não contempla o aspecto, seja em razão de escopo declarado, seja pela ausência de qualquer referência direta ou indireta ao tema.

A aplicação desses critérios exigiu reconhecer que os documentos analisados possuem naturezas distintas, o que influencia o que pode ser considerado como orientação "suficiente" para fins de classificação. Normas de gestão, como a ISO/IEC 27001 (ABNT, 2022), enunciam controles que estabelecem o que deve existir sem detalhar como implementar; *frameworks* orientados a resultados, como o NIST CSF (NIST, 2024), descrevem os resultados esperados sem prescrever mecanismos específicos; guias técnicos, como o OWASP *Secure by Design Framework* (OWASP, 2025b), fornecem orientação de implementação em nível de arquitetura; e documentos de caráter legal, como a LGPD (BRASIL, 2018), estabelecem obrigações sem prescrever controles técnicos. Diante dessa heterogeneidade, o critério de

"orientação suficiente" foi aplicado de acordo com a natureza de cada fonte: para normas e *frameworks* de gestão, considerou-se suficiente o enunciado de um requisito ou resultado claro e nominalmente relacionado ao aspecto; para guias técnicos, exigiu-se a presença de recomendações de implementação concretas; para documentos legais, considerou-se suficiente o estabelecimento de uma obrigação expressamente vinculada ao aspecto.

Essa escala de classificação foi aplicada a todos os documentos e aspectos analisados, com base na leitura de cada fonte. Nos casos em que a classificação apresentou ambiguidade, adotou-se o nível mais conservador, isto é, o menor nível de cobertura entre os dois plausíveis, de forma a evitar superestimar a aderência de uma fonte.

Essa estrutura de categorização permite uma análise mais precisa das convergências, divergências e lacunas entre as diferentes fontes, além de apoiar a consolidação das diretrizes no guia proposto. Os resultados dessa análise comparativa são apresentados e discutidos na seção de Análise e Discussão, sendo posteriormente utilizados como base para a construção do guia de diretrizes de segurança.

2.3 CRITÉRIOS DE PRIORIZAÇÃO

Considerando a diversidade de normas, legislações, *frameworks* e boas práticas analisadas neste trabalho, foi necessário estabelecer critérios objetivos para selecionar e priorizar as diretrizes que compõem o guia proposto. A priorização foi realizada com base em quatro critérios principais: impacto na segurança, aplicabilidade prática, relevância normativa e adequação ao contexto de aplicações governamentais. Cada critério foi definido de forma a considerar tanto a importância técnica quanto sua viabilidade de adoção por equipes de desenvolvimento.

Quadro 1 – Critérios de priorização de diretrizes

Critério	O que avalia	Por que foi selecionado
Impacto na segurança	Potencial de reduzir riscos à confidencialidade, integridade e disponibilidade.	Garante que o guia priorize práticas com contribuição real para a segurança.
Aplicabilidade prática e aderência ao SDLC	Relação com requisitos, projeto, codificação, testes, implantação ou manutenção.	Mantém o guia alinhado ao objetivo do trabalho e evita recomendações abstratas ou apenas gerenciais.
Relevância normativa e recorrência nas fontes	Presença e recorrência em normas, legislações ou <i>frameworks</i> reconhecidos.	Garante embasamento técnico, legal e institucional.
Adequação ao contexto de aplicações governamentais	Relação com riscos e obrigações do setor público	Garante aderência ao escopo de aplicações governamentais.

Fonte: elaborado pela autora

2.3.1 Impacto na segurança

O critério de impacto na segurança foi adotado para avaliar o potencial de cada diretriz em reduzir riscos relevantes à confidencialidade, integridade e disponibilidade das informações. Uma diretriz foi considerada de maior impacto quando sua ausência pudesse favorecer vulnerabilidades críticas, exposição indevida de dados, falhas de autenticação, acesso não autorizado, indisponibilidade do serviço ou comprometimento da rastreabilidade das ações no sistema.

2.3.2 Aplicabilidade prática e aderência ao SDLC

O critério de aplicabilidade prática e aderência ao ciclo de vida de desenvolvimento de *software* (SDLC) foi adotado para avaliar, de forma integrada, se a diretriz pode ser compreendida, executada e incorporada às atividades realizadas por desenvolvedores ao longo do processo de desenvolvimento.

Foram priorizadas diretrizes que pudessem ser traduzidas em ações objetivas dentro das etapas do SDLC, como definição de requisitos de segurança, aplicação de princípios de arquitetura segura, validação de entradas durante a implementação, realização de testes de segurança e configuração adequada na implantação.

Esse critério foi definido a partir da constatação de que muitas normas e *frameworks* apresentam recomendações em nível elevado de abstração, frequentemente voltadas à governança, à gestão organizacional ou ao gerenciamento de riscos. Embora essas recomendações sejam relevantes, nem sempre são diretamente acionáveis no contexto técnico do desenvolvimento de *software*.

2.3.3 Relevância normativa e recorrência nas fontes

O critério de relevância normativa foi utilizado para avaliar o grau de sustentação da diretriz nas fontes analisadas. Foram consideradas mais relevantes as diretrizes presentes em documentos normativos nacionais, legislações aplicáveis ao setor público ou *frameworks* amplamente reconhecidos. Além disso, diretrizes recorrentes em mais de uma fonte receberam maior prioridade, por indicarem consenso entre diferentes abordagens. Esse critério foi selecionado para garantir que o guia não fosse baseado em recomendações isoladas, mas em práticas reconhecidas por fontes consolidadas.

2.3.4 Adequação ao contexto de aplicações governamentais

O critério de adequação ao contexto governamental foi utilizado para avaliar se a diretriz possui relação com riscos, obrigações e características específicas de aplicações utilizadas no setor público. Foram priorizadas diretrizes relacionadas à proteção de dados pessoais, transparência, rastreabilidade, controle de acesso, minimização de dados e conformidade com requisitos legais e institucionais.

Esse critério foi selecionado porque aplicações governamentais frequentemente tratam dados pessoais e sensíveis de cidadãos, exigindo maior atenção à privacidade, à segurança e à responsabilização no tratamento dessas informações. Assim, recomendações diretamente relacionadas ao contexto da Administração Pública Federal foram consideradas mais relevantes para o escopo do trabalho.

2.4 VERIFICAÇÃO DA COBERTURA FRENTE AO OWASP TOP 10

Esta etapa da metodologia consiste em uma verificação de cobertura do guia frente às dez categorias de vulnerabilidade listadas no OWASP Top 10 (OWASP, 2025c). Essa etapa é executada após a consolidação das diretrizes resultantes das etapas anteriores e tem como objetivo identificar se alguma categoria de vulnerabilidade relevante permanece sem endereçamento adequado no guia.

A verificação é conduzida por meio de uma análise por categorias, estruturada em três passos. No primeiro passo, cada uma das dez categorias do Top 10 é analisada individualmente, com base na descrição de vetores de ataque e nas abordagens de mitigação recomendadas pelo próprio documento. No segundo passo, essas informações são confrontadas com o conjunto de diretrizes já consolidadas, verificando se ao menos uma diretriz do guia contribui diretamente para reduzir a probabilidade de ocorrência ou o impacto da vulnerabilidade em questão. Para que uma categoria do Top 10 seja considerada coberta, a diretriz correspondente precisa atuar sobre a causa raiz da vulnerabilidade, e não apenas sobre seus efeitos, conforme descrito na seção de prevenção do documento. No terceiro passo, as categorias identificadas como não-cobertas são registradas, e para cada uma delas são buscadas práticas de mitigação aplicáveis ao contexto de desenvolvimento de aplicações.

Os resultados dessa etapa são apresentados na seção 6, Desenvolvimento do Guia, por meio de uma matriz de cobertura, na qual cada categoria do Top 10 é relacionada à diretriz ou conjunto de diretrizes que a endereça. Nos casos em que houve adição de novas diretrizes a partir dessa etapa, a matriz indica explicitamente essa origem, distinguindo-as das diretrizes provenientes das etapas anteriores. Nos casos em que uma lacuna identificada não gerou inclusão de novas diretrizes por não atender aos critérios de priorização, isso também é registrado na matriz, com a justificativa correspondente.

2.5 AVALIAÇÃO DO GUIA

Esta etapa tem como objetivo verificar se as diretrizes consolidadas ao longo do trabalho oferecem cobertura adequada frente a requisitos técnicos de segurança reconhecidos internacionalmente, avaliando a consistência e a suficiência do guia proposto em relação a um padrão independente do processo de construção.

Para a avaliação do guia proposto, foi selecionado o OWASP *Application Security Verification Standard* (ASVS) como instrumento de referência. A escolha se justifica por quatro razões principais:

1. O ASVS não foi utilizado em nenhuma das etapas de construção do guia, o que garante a independência necessária para que a verificação não seja circular.
2. Diferentemente das demais fontes analisadas neste trabalho, que operam em nível gerencial ou descrevem práticas de implementação, o ASVS define requisitos técnicos específicos e verificáveis para aplicações, organizados por nível de maturidade, o que permite um mapeamento objetivo entre suas exigências e as diretrizes propostas.
3. Seu escopo é inteiramente voltado à segurança de aplicações, alinhando-se diretamente ao foco do guia.
4. O documento possui reconhecimento internacional equivalente ao das fontes utilizadas na construção, conferindo autoridade à avaliação realizada.

Outras alternativas foram consideradas e descartadas: a ISO 27001 e o NIST CSF já foram utilizados como fontes de construção e têm foco organizacional, não técnico. O OWASP Top 10 foi utilizado na etapa de verificação de lacunas durante a construção, e a validação por especialistas não foi viável dentro do escopo e prazo deste trabalho, ficando como sugestão de continuidade em trabalhos futuros.

A avaliação é conduzida por meio de um mapeamento sistemático entre as diretrizes do guia e os requisitos do ASVS. O processo segue três passos. No primeiro passo, os requisitos do ASVS são filtrados pelo nível de maturidade mais básico, o Nível 1, que corresponde ao conjunto mínimo de controles recomendados para qualquer aplicação, independentemente do contexto. Esse recorte é adotado porque o guia se destina a equipes de desenvolvimento com maturidade variável em segurança, e exigir cobertura de níveis mais avançados extrapolaria o escopo deste trabalho. No segundo passo, cada requisito do ASVS selecionado é confrontado com o conjunto de diretrizes do guia, verificando se ao menos uma diretriz o endereça de forma direta ou equivalente. No terceiro passo, os resultados são registrados em uma matriz de correspondência, na qual cada requisito é classificado como coberto, parcialmente coberto ou não coberto, com indicação da diretriz correspondente nos dois primeiros casos e com registro justificado no terceiro.

3 FUNDAMENTAÇÃO TEÓRICA

O objetivo desta seção é apresentar a base conceitual necessária para a compreensão dos temas que fundamentam este trabalho, explorando aspectos técnicos e também elementos de natureza mais abrangente. A inclusão desses tópicos busca contextualizar componentes do desenvolvimento de *software*, oferecendo sustentação teórica para as análises e discussões apresentadas ao longo do estudo.

3.1 FRAMEWORK

Um *framework* é um conjunto organizado e reutilizável de componentes, bibliotecas, e boas práticas que fornece uma estrutura e orientação para desenvolver um software (AWS, 2025). Ele pode definir a arquitetura, padrões de design, ciclos de vida e fornece serviços comuns (ex.: roteamento, persistência, injeção de dependência). Exemplos de *frameworks* conhecidos: Spring, Django, React.

Frameworks não se limitam a ferramentas de código, eles também podem assumir a forma de modelos conceituais e normativas, que proporcionam uma estrutura de princípios, diretrizes, processos e convenções para a tomada de decisões mais técnicas (AWS, 2025). A exemplo disso, neste trabalho utilizamos o OWASP *Secure by Design Framework* e o NIST *Cybersecurity Framework*. Ambos estabelecem estruturas organizadas de princípios, práticas recomendadas e processos padronizados para orientar equipes de desenvolvimento e segurança na construção e manutenção de sistemas confiáveis.

Como já citado, o objetivo de um *framework* é estabelecer um modelo completo e estruturado de execução ou governança, definindo processos, artefatos, responsabilidades e métricas que devem ser adotados de forma integrada ao ciclo de vida de desenvolvimento. No entanto, esse nível de formalização ultrapassa o escopo definido para este trabalho e não atende diretamente à principal necessidade identificada: a falta de orientações claras, práticas e acessíveis para que desenvolvedores possam aplicar os requisitos e diretrizes de segurança em seus contextos específicos, independentemente do modelo de desenvolvimento adotado. Diante disso, opta-se pela elaboração de um guia de diretrizes e boas práticas, que

permite consolidar e traduzir as exigências normativas de maneira flexível e aplicável, sem impor uma estrutura metodológica como a de um *framework*.

3.2 SEGURANÇA DE REDES, CIBERSEGURANÇA E SEGURANÇA DA INFORMAÇÃO

Cibersegurança é um campo amplo que aborda proteção contra ameaças digitais ao longo de todo o ambiente computacional: redes, *endpoints*, aplicações, *cloud*, identidade, operações, entre outros. Inclui práticas de governança, resposta a incidentes e inteligência de ameaças (SOMMERVILLE, 2019).

Segurança de redes é um componente técnico especializado dentro da cibersegurança que tem como objetivo proteger a conectividade e blindar vetores que transitam pela rede, focando na proteção da infraestrutura e dos dados em movimento nesta camada (SOMMERVILLE, 2019). Exemplos de temas relacionados à segurança de redes: *firewalls*, segmentação, VPNs (*Virtual Private Networks*), controle de acesso a *switches/routers*, mitigação DDoS (*Distributed Denial of Service*), criptografia de tráfego.

Segurança da informação é uma área gerencial que tem como foco proteger a confidencialidade, integridade e disponibilidade das informações, independente do suporte (físico, digital, humano). Abrange políticas, classificação de dados, controles administrativos, treinamento e conformidade legal (ex.: LGPD) (BRASIL, 2018). Segurança de informação é sobre gestão de risco e controles, enquanto cibersegurança e segurança de redes são subconjuntos operacionais dentro dela (SOMMERVILLE, 2019).

A delimitação deste trabalho para o campo da segurança da informação decorre da necessidade de concentrar a análise nos princípios, controles e práticas que impactam diretamente o desenvolvimento de aplicações utilizadas pelo Governo. Embora segurança de redes e cibersegurança sejam áreas fundamentais dentro do ecossistema de proteção digital, ambas possuem foco predominantemente infraestrutural e operacional, enquanto a segurança da informação possui uma abordagem mais abrangente, estratégica e orientada à proteção dos dados e dos processos, elementos centrais para o desenvolvimento seguro de *software*.

A segurança de redes está voltada sobretudo à proteção dos meios de comunicação, do tráfego de dados e dos dispositivos conectados, abordando temas

como *firewalls*, segmentação, VPNs e controle de tráfego. Apesar de essenciais para o ambiente onde uma aplicação opera, esses controles não dependem diretamente das atividades do desenvolvedor, e sim da equipe de infraestrutura ou operações de TI. Portanto, tratá-los de forma aprofundada fugiria ao foco do trabalho, cujo objetivo é orientar práticas que os desenvolvedores podem aplicar diretamente no código, na arquitetura e nas decisões de projeto.

A cibersegurança, por sua vez, envolve práticas mais amplas relacionadas à defesa dos sistemas contra ameaças digitais (SOMMERVILLE, 2019), incluindo monitoramento, resposta a incidentes, análise de *malware*, proteção de *endpoints* e operações de segurança. Esses elementos, embora relevantes para o ciclo de vida de um sistema, também se situam prioritariamente no âmbito operacional e estratégico das organizações, não compondo o núcleo de responsabilidades do desenvolvedor de aplicações.

Já a segurança da informação, estruturada sobre os princípios de confidencialidade, integridade e disponibilidade, abrange controles administrativos, técnicos e processuais diretamente relacionados ao tratamento seguro de dados e ao desenvolvimento seguro de *software*. É nesse domínio que se inserem temas como requisitos de segurança, políticas de acesso, classificação e proteção de dados, segurança por *design*, criptografia aplicada, validações, autenticação, autorização e prevenção de vulnerabilidades, todas áreas nas quais o desenvolvedor exerce influência direta e imediata.

3.3 CONFIDENCIALIDADE, INTEGRIDADE E DISPONIBILIDADE

A segurança da informação apoia-se tradicionalmente na tríade formada pelos princípios de confidencialidade, integridade e disponibilidade, que representam os principais objetivos a serem preservados no tratamento de dados e sistemas. Esses princípios orientam tanto práticas técnicas quanto diretrizes organizacionais, servindo como base conceitual para políticas, controles e mecanismos de proteção.

3.3.1 Confidencialidade

Refere-se à garantia de que a informação esteja acessível apenas a indivíduos, processos ou sistemas devidamente autorizados. Envolve impedir a

divulgação não autorizada de dados e proteger informações sensíveis, incluindo aspectos relacionados à privacidade dos titulares. A violação da confidencialidade ocorre quando informações são expostas, acessadas ou compartilhadas sem permissão (STALLINGS, 2015).

Para garantir a confidencialidade, diversas medidas podem ser empregadas, como criptografia de dados em repouso e em trânsito, autenticação forte, controle de acesso baseado em privilégios mínimos, segmentação de redes e uso de canais seguros de comunicação. Esses mecanismos reduzem a superfície de exposição e asseguram que apenas entidades autorizadas possam manipular ou visualizar informações sensíveis.

3.3.2 Integridade

Diz respeito à precisão, completude e confiabilidade da informação ao longo de todo o seu ciclo de vida. Esse princípio busca evitar modificações indevidas, corrupções, destruição não autorizada ou manipulações maliciosas dos dados. A integridade também abrange conceitos como autenticidade e irretratabilidade, assegurando que a informação seja verdadeira, verificável e que ações não possam ser negadas posteriormente (STALLINGS, 2015).

A garantia da integridade envolve o uso de mecanismos como assinaturas digitais, códigos de verificação (*hashes*), trilhas de auditoria, controles de versão, validação de entrada e monitoramento contínuo. Esses controles permitem identificar mudanças não autorizadas, verificar se os dados permanecem íntegros e assegurar que operações críticas possam ser rastreadas e autenticadas.

3.3.3 Disponibilidade

Consiste em garantir que as informações, serviços e sistemas estejam acessíveis e utilizáveis sempre que necessários por usuários autorizados. Esse princípio envolve assegurar continuidade operacional, minimizar indisponibilidades e prover respostas rápidas, especialmente em ambientes que exigem alto desempenho e confiabilidade (STALLINGS, 2015).

A disponibilidade pode ser assegurada por meio de redundância de infraestrutura, balanceamento de carga, uso de servidores espelhados, *backups* frequentes, mecanismos de tolerância a falhas e planos de recuperação de desastres. Além disso, práticas como monitoração proativa, mitigação de ataques de negação de serviço (DDoS) e atualizações regulares garantem que o sistema permaneça operacional mesmo em condições adversas ou diante de falhas inesperadas.

3.4 DESENVOLVIMENTO DE SOFTWARE

O desenvolvimento de *software* compreende o conjunto estruturado de atividades, práticas e decisões que orientam a criação, evolução e manutenção de sistemas computacionais (SOMMERVILLE, 2019). A compreensão desse processo é fundamental para este trabalho, uma vez que aspectos relacionados ao tema influenciam diretamente a qualidade, a segurança e a conformidade das aplicações.

3.4.1 Ciclo de vida de desenvolvimento de *software*

O ciclo de vida de desenvolvimento de *software* (do inglês, *Software Development Life Cycle*, SDLC) é uma representação simplificada de um processo de *software*, que descreve as atividades envolvidas no desenvolvimento e a sequência em que essas atividades ocorrem (SOMMERVILLE, 2019). Cada modelo de SDLC representa o processo a partir de uma perspectiva particular, fornecendo informações parciais sobre ele. Um modelo focado em atividades, por exemplo, não necessariamente descreve os papéis das pessoas envolvidas em cada etapa.

Um dos modelos mais tradicionais associados ao SDLC é o modelo cascata (*waterfall*), que organiza o desenvolvimento em fases sequenciais e distintas. Na prática, porém, essas etapas se sobrepõem, pois “O processo de *software*, na prática, nunca é um modelo linear simples, pois envolve *feedback* entre as fases” (SOMMERVILLE, 2019, p. 34). Os estágios clássicos que compõem o modelo cascata incluem:

1. Análise e definição de requisitos
2. Projeto do sistema
3. Implementação (codificação)
4. Integração e testes

5. Operação e manutenção

Cada uma dessas fases possui um propósito distinto no processo de construção do *software*. Na análise e definição de requisitos, as necessidades dos usuários e partes interessadas são identificadas, analisadas e documentadas, resultando na especificação do que o sistema deverá fazer, tanto em termos funcionais quanto em termos de atributos de qualidade, como desempenho e segurança. Na fase de projeto, os requisitos levantados são traduzidos em decisões arquiteturais e técnicas, definindo como o sistema será estruturado internamente. A implementação corresponde à escrita do código-fonte com base no projeto definido. Os testes verificam se o *software* produzido atende aos requisitos especificados e identificam defeitos a serem corrigidos. Por fim, a operação e manutenção cobre o período após a entrega do sistema, no qual ele é monitorado, atualizado e corrigido conforme necessário (SOMMERVILLE, 2019).

Sommerville (2019) observa que, na prática, a maior parte dos processos de *software* se baseia em um modelo genérico, mas frequentemente incorpora características de outros modelos. Abordagens iterativas e incrementais, como os métodos ágeis, organizam essas mesmas atividades em ciclos menores e repetidos, revisitando requisitos, projeto e testes a cada iteração. Abordagens como DevSecOps, por sua vez, integram práticas de segurança e qualidade de forma contínua ao longo do pipeline de entrega, automatizando verificações que no modelo cascata ocorreriam em fases separadas. Independentemente da abordagem adotada, as atividades essenciais do SDLC (levantar requisitos, projetar, implementar, testar e manutenção) estão presentes em qualquer processo de desenvolvimento, ainda que organizadas de formas distintas.

Por essa razão, e considerando que o guia proposto neste trabalho é destinado a equipes com diferentes níveis de maturidade técnica e diferentes processos de desenvolvimento, as fases do SDLC serão utilizadas como estrutura organizacional para a apresentação das diretrizes de segurança. Essa escolha garante que o guia seja acessível e aplicável sem exigir mudanças no processo já adotado pela equipe. Cada diretriz pode ser aplicada no momento em que a atividade correspondente ocorre dentro do processo de desenvolvimento em uso, independentemente de como esteja denominado ou estruturado.

3.4.2 Ciclo de vida de desenvolvimento de *software* seguro

O Ciclo de Vida de Desenvolvimento de *Software* Seguro (do inglês, *Secure Software Development Life Cycle*, S-SDLC) é uma evolução do SDLC tradicional que incorpora práticas de segurança da informação em cada uma de suas fases, de forma sistemática e integrada. Seu objetivo central é tratar a segurança não como uma etapa isolada ou corretiva ao final do processo, mas como um atributo de qualidade presente desde a concepção do *software* até a sua manutenção contínua (NAZIR; NAZIR, 2018).

A principal diferença em relação ao SDLC está no que é exigido do desenvolvedor em cada fase. Karim et al. (2016), a partir de investigação em organizações de desenvolvimento de *software*, identificaram que na etapa de especificação de requisitos, além de identificar as necessidades funcionais dos usuários, o S-SDLC demanda que requisitos de segurança sejam levantados simultaneamente. Na fase de projeto, são realizadas análises de ameaças e os controles de segurança são incorporados à arquitetura do sistema. Durante a codificação, as melhores práticas de desenvolvimento seguro devem ser seguidas para minimizar a introdução de vulnerabilidades. Na fase de testes, o *software* é submetido a verificações específicas de segurança, e não apenas à validação funcional. Por fim, na implantação e manutenção, são aplicados regularmente *patches* de segurança e procedimentos de resposta a incidentes.

Conforme apontam Nazir e Nazir (2018), a adoção do S-SDLC traz vantagens relevantes em relação ao modelo tradicional, entre elas a mitigação antecipada de riscos de segurança, a redução de custos com correções tardias e o auxílio à conformidade regulatória. No entanto, o modelo também apresenta limitações práticas: requer investimento inicial mais elevado, demanda conhecimentos especializados em segurança que nem sempre estão disponíveis nas equipes de desenvolvimento, e pode aumentar o tempo total do processo quando mal dimensionado. Essas características explicam, em partes, o fato de que o S-SDLC raramente é adotado de forma integral na prática. Profissionais tendem a selecionar apenas as atividades consideradas mais adequadas ao seu contexto de trabalho, especialmente em ambientes ágeis (RINDELL et al., 2021).

Por fim, é importante destacar que tanto o SDLC quanto o S-SDLC foram originalmente concebidos como modelos de fases sequenciais, o que nem sempre

reflete a realidade de equipes que adotam abordagens iterativas e incrementais de desenvolvimento. Nesse tipo de contexto, as práticas de segurança precisam ser revisitadas e aplicadas de forma contínua a cada incremento, e não apenas uma vez ao longo do projeto. Diante disso, este trabalho adota o SDLC como estrutura organizacional para a apresentação das diretrizes de segurança propostas no guia, enquanto o S-SDLC é utilizado como orientação conceitual do guia: integrar segurança ao desenvolvimento desde o início de forma planejada, e não como um acréscimo posterior.

3.4.3 Segurança por *design*

O conceito de Segurança por *Design* (*Security by Design*) estabelece que mecanismos, princípios e práticas de segurança devem ser incorporados ao *software* desde as fases iniciais de concepção e arquitetura, e não adicionados apenas como etapas posteriores ou corretivas (OWASP, 2025b). Essa abordagem reconhece que a segurança não é um componente opcional, mas um atributo fundamental de qualidade, que precisa ser planejado, projetado e implementado de forma sistemática.

Segundo Deogun et al. (2019), a segurança deve ser tratada como parte integrante do processo de engenharia de *software*, orientando decisões desde a definição de requisitos, passando pelo desenho da arquitetura, pela modelagem de ameaças e pelas estratégias de mitigação, até a implementação e a operação contínua. Os autores argumentam que, quando a segurança é considerada apenas nas etapas finais, ou após incidentes, o resultado tende a ser mais custoso, menos eficaz e frequentemente insuficiente para mitigar riscos de forma abrangente.

Segurança por *Design* é mais do que um conjunto de boas práticas: trata-se de um paradigma de desenvolvimento que busca garantir que sistemas sejam concebidos com segurança inerente, reduzindo vulnerabilidades, antecipando riscos e promovendo soluções que protejam dados e usuários ao longo de todo o ciclo de vida do *software*.

3.4.4 Requisitos funcionais e não funcionais

No contexto de engenharia de *software*, os requisitos representam as características, comportamentos e condições que um sistema deve atender para cumprir seus objetivos (SOMMERVILLE, 2019). De modo geral, eles se dividem em requisitos funcionais e requisitos não funcionais, categorias que desempenham papéis complementares no processo de desenvolvimento.

Requisitos funcionais descrevem a funcionalidades de um sistema, o que ele deve fazer, especificando serviços, operações e comportamentos esperados. Eles definem as ações diretamente observáveis pelo usuário ou por outros sistemas. Devem ser claramente definidos para evitar qualquer falha de compreensão e alinhamento entre o que é desenvolvido em relação a expectativa dos usuários finais (OKESOLA *et al.*, 2020). Em essência, tratam do comportamento explícito da aplicação e são essenciais para garantir que o *software* atenda às necessidades operacionais para as quais foi concebido.

Por outro lado, requisitos não funcionais tratam de como o sistema deve operar, estabelecendo critérios de qualidade, restrições e atributos que influenciam a experiência, eficiência, compatibilidade, resiliência e confiabilidade do software (SOMMERVILLE, 2019). Eles não se referem a funcionalidades específicas, mas às propriedades globais do sistema, como desempenho, escalabilidade, usabilidade, segurança, disponibilidade e conformidade regulatória. Esses requisitos são fundamentais para garantir que a aplicação funcione de maneira adequada em diferentes contextos e sob condições variadas, influenciando diretamente a robustez, a qualidade e a maturidade do produto final (BALOGUN, 2018).

Assim, enquanto os requisitos funcionais definem o conjunto de funcionalidades do software, os requisitos não funcionais determinam o nível de qualidade com que essas funcionalidades devem ser entregues. Ambos são indispensáveis para um desenvolvimento consistente e alinhado às expectativas dos usuários e às necessidades organizacionais.

3.5 DADOS E A LEI GERAL DE PROTEÇÃO DE DADOS PESSOAIS

É essencial compreender que a proteção eficaz das informações depende também do entendimento sobre os tipos de dados tratados pelos sistemas. No contexto de aplicações que coletam, armazenam e processam informações sensíveis dos usuários, conhecer essas categorias é parte fundamental da própria construção

de segurança. A Lei Geral de Proteção de Dados Pessoais (LGPD) estabelece categorias distintas de dados, cada uma com exigências específicas quanto ao tratamento, armazenamento e proteção. A forma como cada categoria é manipulada afeta diretamente o nível de risco envolvido e as obrigações legais do controlador.

Além disso, a LGPD estabelece que os direitos do titular, como o fornecimento e a revogação do consentimento, a possibilidade de solicitar eliminação de dados e a revisão de decisões automatizadas, também variam conforme o tipo de dado tratado. Isso exige que as aplicações incorporem mecanismos que permitam ao usuário exercer controle sobre suas informações, incluindo interfaces para gerenciar permissões, apagar dados armazenados e revisar autorizações previamente concedidas. Dessa forma, compreender as categorias de dados está diretamente relacionado à implementação de funcionalidades e controles que suportem um tratamento adequado e seguro.

A seguir, apresentam-se as principais categorias de dados consideradas neste trabalho:

3.5.1 Dados Pessoais

Dados pessoais são informações que permitem identificar, direta ou indiretamente, uma pessoa (BRASIL, Lei nº 13.709/2018). Exemplos incluem nome, CPF, RG, endereço, e-mail, telefone, entre outros. Durante o desenvolvimento de aplicações, esses dados demandam práticas básicas de proteção, autenticação, controles de acesso e armazenamento adequado por comporem a base de identificação do usuário e poderem ser utilizados de forma maliciosa em fraudes, engenharia social e roubo de identidade.

3.5.2 Dados Pessoais Sensíveis

Dados sensíveis são informações que revelam aspectos íntimos, identificadores especiais ou elementos que podem causar discriminação ao titular (BRASIL, Lei nº 13.709/2018). A LGPD classifica como sensíveis dados sobre:

1. origem racial ou étnica
2. convicção religiosa

3. opinião política
4. saúde, assistência social ou vida sexual
5. dados genéticos ou biométricos
6. dados de localização
7. dados referentes a crianças, adolescentes e grupos vulneráveis

Por seu alto risco, esses dados exigem camadas reforçadas de segurança, incluindo criptografia forte, políticas restritivas de acesso, minimização de coleta e mecanismos adicionais de consentimento. No contexto governamental, muitas aplicações realizam tratamento de informações extremamente sensíveis, como prontuários médicos, dados previdenciários e biometria, ampliando a necessidade de rigor técnico.

3.5.3 Dados Públicos

São dados cuja divulgação é permitida por lei ou que foram tornados públicos pelo próprio titular. Mesmo assim, a LGPD estabelece que o tratamento deve observar a finalidade original e evitar uso abusivo (BRASIL, Lei nº 13.709/2018).

No desenvolvimento de aplicações, dados públicos não dispensam controles de segurança, já que seu tratamento inadequado ainda pode gerar riscos, sobretudo quando combinados com informações privadas.

3.6 TRANSPORT LAYER SECURITY (TLS)

O *Transport Layer Security* (TLS) é um protocolo criptográfico que estabelece um canal seguro entre dois pontos de comunicação em uma rede (cliente e servidor) garantindo confidencialidade, integridade e autenticidade dos dados transmitidos (STALLINGS, 2015). Seu funcionamento se baseia em um processo de negociação inicial (*handshake*), no qual as partes acordam a versão do protocolo e os algoritmos criptográficos a serem utilizados, trocam certificados digitais para verificação de identidade e estabelecem as chaves da sessão cifrada.

A ausência ou má configuração do TLS expõe a comunicação a ataques do tipo *man-in-the-middle*, nos quais um agente malicioso intercepta ou modifica o tráfego sem que nenhuma das partes perceba. O PPSI (2023) reforça que dados sensíveis que trafegam por canais compartilhados, como redes *Wi-Fi* e redes celulares, podem

ser interceptados e modificados, recomendando que as aplicações imponham o uso de canal seguro de ponta a ponta ao transmitir qualquer informação sensível, incluindo credenciais e *tokens* de autenticação.

Habilitar o TLS, porém, não é suficiente por si só. O OWASP ASVS v5.0.0 (OWASP, 2025a) estabelece que somente as versões mais recentes (1.2+) do protocolo devem ser habilitadas e que certificados devem ser emitidos por Autoridades Certificadoras confiáveis. O OWASP *Secure Coding Practices* (OWASP, 2022) complementa ao determinar que, em caso de falha na conexão TLS, o sistema não deve oferecer alternativa de conexão insegura.

No contexto governamental, o uso correto do TLS extrapola a boa prática técnica e torna-se obrigação legal: o art. 46 da LGPD (BRASIL, 2018) exige que os agentes de tratamento adotem medidas técnicas aptas a proteger dados pessoais de acessos não autorizados e de situações ilícitas de comunicação ou difusão, o que torna a configuração adequada do protocolo um requisito mínimo para qualquer aplicação que processe dados de cidadãos.

3.7 MODELAGEM DE AMEAÇAS (THREAT MODELING)

A modelagem de ameaças (em inglês, *threat modeling*) é uma prática que consiste em identificar, analisar e priorizar potenciais ameaças a um sistema ainda nas fases iniciais de seu desenvolvimento, antes de iniciar a codificação. Seu objetivo central é antecipar como um agente malicioso poderia comprometer o sistema, permitindo que contramedidas sejam incorporadas ao projeto e não apenas corrigidas após o fato (OWASP, 2025b).

Entre as técnicas mais utilizadas para conduzir uma modelagem de ameaças está o STRIDE, que categoriza as ameaças em seis tipos: falsificação de identidade (*Spoofing*), adulteração de dados (*Tampering*), repúdio (*Repudiation*), divulgação de informações (*Information Disclosure*), negação de serviço (*Denial of Service*) e elevação de privilégios (*Elevation of Privilege*). Por meio dessa taxonomia, é possível percorrer os fluxos de dados e componentes do sistema avaliando cada categoria de ameaça de forma sistemática, o que resulta em um registro estruturado de riscos e das decisões de mitigação tomadas (OWASP, 2025b).

No contexto governamental, o Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024) contextualiza a identificação e o tratamento

de riscos como atividades essenciais para os órgãos da Administração Pública Federal, alinhando essa prática às obrigações de gestão de riscos estabelecidas pelos normativos do GSI/PR. aplicações governamentais processam dados pessoais sensíveis de cidadãos em larga escala, o que amplia o impacto potencial de uma falha de *design*.

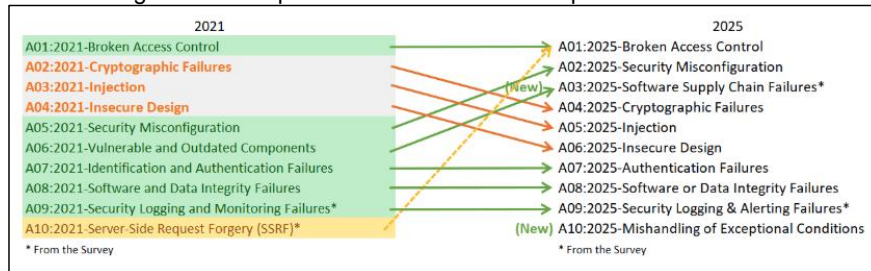
3.8 OWASP TOP 10

O *Open Web Application Security Project* (OWASP) é uma fundação internacional sem fins lucrativos dedicada à melhoria da segurança de aplicações. Seus projetos, ferramentas e materiais são de acesso livre e desenvolvidos de forma colaborativa por pesquisadores, profissionais e voluntários ao redor do mundo (OWASP, 2022). Entre as publicações mantidas pela organização, o OWASP Top 10 se destaca como uma das referências adotadas na área de segurança de aplicações em nível global.

O OWASP Top 10 é um documento que lista as dez categorias de vulnerabilidades mais recorrentes em aplicações, construído a partir de dados reais coletados junto a centenas de organizações. Publicado pela primeira vez em 2003 e atualizado periodicamente, o documento tornou-se referência de segurança para equipes de desenvolvimento, sendo utilizado como base para programas de treinamento, políticas de desenvolvimento seguro e auditorias de conformidade em organizações.

É importante compreender que o Top 10 não elenca necessariamente as vulnerabilidades de maior impacto técnico, mas sim as de maior frequência de ocorrência em aplicações reais. Como mostra a Figura 1, a lista vai sofrendo alterações ao longo do tempo: algumas vulnerabilidades são removidas, outras adicionadas, ou mudam de posição no *ranking*, o que reflete a evolução das práticas de ataque e das tecnologias utilizadas no desenvolvimento. A versão mais recente, publicada em 2025, organiza as vulnerabilidades nas seguintes categorias: Quebra de Controle de Acesso (A01), Configuração de Segurança Incorreta (A02), Falhas na Cadeia de Suprimentos de *Software* (A03), Falhas Criptográficas (A04), Injeção (A05), *Design* Inseguro (A06), Falhas de Autenticação (A07), Falhas na Integridade de *Software* ou Dados (A08), Falhas de Registro e Monitoramento de Segurança (A09) e Tratamento Inadequado de Exceções (A10) (OWASP, 2025c).

Figura 1 – Comparativo entre o OWASP Top 10 de 2021 e 2025.



Fonte: OWASP (2025c)

Diferentemente de guias de boas práticas, que organizam suas recomendações em termos de controles e ações a serem adotadas, o Top 10 parte da perspectiva da ameaça: para cada categoria, descreve como a vulnerabilidade ocorre, quais condições a favorecem e quais abordagens são eficazes para sua prevenção. De acordo com o próprio OWASP, a maioria das vulnerabilidades listadas ocorre durante a etapa de desenvolvimento da aplicação, e a recomendação é que as medidas de segurança sejam observadas ainda nessa fase, de forma que as aplicações não sejam disponibilizadas em produção com tais vulnerabilidades presentes (OWASP, 2025c).

Essa perspectiva orientada à vulnerabilidade complementa a abordagem orientada à prática adotada por documentos como o OWASP *Secure Coding Practices* e o OWASP *Secure by Design Framework*, desenvolvidos, também, para traduzir o conhecimento sobre vulnerabilidades catalogadas no Top 10 em práticas concretas de desenvolvimento. No contexto deste trabalho, o Top 10 desempenha um papel metodológico específico, descrito na seção 2.4: após a consolidação das diretrizes por meio das etapas de análise comparativa e priorização, o documento é utilizado como instrumento de verificação de cobertura, permitindo identificar se alguma categoria de vulnerabilidade relevante permanece sem endereçamento adequado no guia proposto.

4 REVISÃO DA LITERATURA

A revisão da literatura realizada para este trabalho busca estabelecer uma ponte entre as diretrizes oficiais do Governo Federal e as melhores práticas utilizadas por empresas privadas e organizações internacionais, de modo a garantir que o guia proposto seja ao mesmo tempo conforme com às exigências legais já existentes e tecnicamente moderno/atualizado.

Os critérios descritos a seguir aplicam-se exclusivamente à seleção das fontes utilizadas na construção do guia. O instrumento selecionado para a etapa de verificação (o OWASP ASVS) obedece a critérios distintos, descritos na seção 2.5, uma vez que seu papel no trabalho é diferente do das demais fontes.

4.1 METODOLOGIA DE SELEÇÃO DAS LITERATURAS

A metodologia adotada para a seleção da literatura baseou-se em buscas realizadas no Google Scholar, visando identificar trabalhos acadêmicos alinhados ao tema deste trabalho. A partir das referências bibliográficas encontradas nesses estudos, foram exploradas outras fontes complementares que apresentassem relação direta com a temática proposta. A escolha dos documentos considerou critérios de relevância, atualidade e aderência ao escopo do trabalho.

O processo de revisão da literatura foi estruturado em três etapas principais:

4.1.1 Identificação de documentos normativos e regulatórios nacionais

Guias, legislações e manuais oficiais publicados por órgãos governamentais, como a LGPD, o Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis, Boas Práticas em Segurança da Informação (TCU) e o Guia do *Framework* de Privacidade e Segurança da Informação do Programa de Privacidade e Segurança da Informação (PPSI) do Gabinete de Segurança da Informação (GSI/PR).

O objetivo desta etapa é compreender as exigências legais e regulatórias aplicáveis ao desenvolvimento de sistemas e aplicações governamentais, de modo a garantir conformidade e segurança jurídica.

4.1.2 Levantamento de boas práticas e frameworks de segurança utilizados no setor privado e em organismos internacionais

Foram utilizadas referências como a *OWASP Secure by Design Framework*, a *ISO/IEC 27001* e a *NIST Cybersecurity Framework*, além de publicações técnicas de empresas de tecnologia reconhecidas no mercado, como IBM e Microsoft.

O objetivo desta etapa é incorporar ao guia soluções práticas, linguagem acessível e exemplos orientados a desenvolvedores, de modo a aproximar a teoria da realidade cotidiana de equipes de desenvolvimento.

4.1.3 Trabalhos acadêmicos relacionados

Nesta etapa, foram selecionados estudos com temas correlatos ao desenvolvimento seguro de *software*, que contribuem para o embasamento teórico e metodológico da pesquisa.

O objetivo desta etapa é utilizar trabalhos acadêmicos consolidados como ponto de partida para a contextualização do tema e também como porta de entrada para novas referências complementares da literatura, fortalecendo a fundamentação teórica do estudo.

4.2 DOCUMENTOS NORMATIVOS E REGULATÓRIOS NACIONAIS

Nesta seção serão detalhados os documentos normativos e regulatórios nacionais selecionados para o desenvolvimento deste trabalho.

4.2.1 Lei geral de proteção de dados pessoais (Lei nº 13.709/2018)

No contexto do desenvolvimento seguro de software, a LGPD impõe que aplicações sejam projetadas considerando os princípios de privacidade desde a concepção (*privacy by design*) e por padrão (*privacy by default*), previstos em seu artigo 46 (BRASIL, Lei nº 13.709/2018). Ou seja, deve-se incorporar requisitos de segurança e privacidade desde as etapas iniciais do ciclo de desenvolvimento, garantindo tratamento dos dados pessoais. Assim, a LGPD serve como referência normativa central para este trabalho, orientando a criação de diretrizes técnicas e

organizacionais como controle de acesso, criptografia e consentimento do usuário, que asseguram conformidade legal e fortalecem a proteção de dados dentro do ciclo de vida do software.

4.2.2 Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (2023)

O documento foi escolhido pela relevância e referência às práticas adotadas pelo Governo Federal, complementando as bases normativas para o desenvolvimento deste TCC. O texto destaca que os aplicativos governamentais devem minimizar a coleta de dados pessoais, restringindo permissões apenas ao estritamente necessário e evitando solicitações excessivas de acesso. Além disso, enfatiza que “...mesmo decisões arquitetônicas aparentemente pequenas podem ter uma grande influência na privacidade e proteção de dados pessoais” e defende a estratégia de design de privacidade que “...envolve a arquitetura de um sistema de informação que compreenda essa temática desde a sua concepção” (PPSI, 2023). Por fim, recomenda que os desenvolvedores implementem mecanismos de controle, imposição e demonstração de conformidade, incluindo registros de atividades, auditorias, relatórios e políticas internas que garantam a responsabilização no tratamento de dados.

4.2.3 Boas Práticas em Segurança da Informação (TCU, 2012)

A obra *Boas Práticas em Segurança da Informação*, publicada pelo Tribunal de Contas da União, fornece uma síntese clara, objetiva e oficialmente fundamentada dos principais tópicos de segurança da informação aplicáveis à Administração Pública Federal. O documento apresenta orientações práticas sobre políticas de segurança, controles de acesso, gestão de ativos, continuidade de negócios e conformidade (TCU, 2012), além de alinhar suas recomendações às normas vigentes, como a ISO/IEC 27002 (ISO/IEC 27002, 2022), e às diretrizes do Gabinete de Segurança Institucional da Presidência da República (GSI/PR).

4.2.4 Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)

O *Guia do Framework de Privacidade e Segurança da Informação* (PPSI, 2024), publicado pelo Gabinete de Segurança Institucional da Presidência da República (GSI/PR), apresenta diretrizes e recomendações voltadas à gestão integrada da privacidade e da segurança da informação em órgãos e entidades públicas. O documento estabelece princípios, responsabilidades e controles técnicos e administrativos necessários para assegurar o tratamento adequado das informações, a continuidade dos serviços e a proteção contra incidentes de segurança. Além de servir como referência normativa, o PPSI propõe uma abordagem estruturada para o gerenciamento de riscos, a classificação de dados e a adoção de medidas de segurança compatíveis com o grau de sensibilidade da informação.

4.3 FRAMEWORKS E BOAS PRÁTICAS DO SETOR PRIVADO E INTERNACIONAL

Nesta seção serão introduzidos padrões e recomendações do setor privado e internacional, que refletem práticas consolidadas e amplamente testadas em diferentes contextos, oferecendo maior maturidade técnica, complementando as diretrizes nacionais do item anterior, e permitindo uma abordagem mais abrangente e robusta.

4.3.1 OWASP Secure by Design Framework (2025)

O *OWASP Secure by Design Framework* apresenta um conjunto de princípios amplamente reconhecidos e consolidados internacionalmente para a construção de software seguro desde as fases iniciais do desenvolvimento (OWASP, 2025b). Diferentemente de manuais focados somente em testes ou correções, o framework enfatiza práticas preventivas, arquitetura segura, redução de superfícies de ataque, proteção contra falhas comuns e abordagem centrada no desenvolvedor. Por essa razão, ele fornece diretrizes modernas e objetivas, tornando-se especialmente relevante para orientar equipes de desenvolvimento que precisam integrar segurança de forma natural e contínua ao ciclo de vida das aplicações.

Além disso, por ser uma iniciativa aberta, mantida por especialistas em segurança de diversas organizações e atualizada conforme novas ameaças emergem, também oferece uma base confiável e prática que complementa as normas e regulamentações governamentais.

4.3.2 ISO/IEC 27001 (2023)

A ISO/IEC 27001 fornece diretrizes que orientam a incorporação de práticas seguras ao longo de todo o ciclo de vida do sistema, incluindo gestão de acessos, controle de mudanças, tratamento de incidentes e proteção de ativos de informação. Para este trabalho, a norma é utilizada como referência para assegurar que o software esteja em conformidade com os padrões internacionais exigidos para certificação em segurança da informação, servindo como base normativa para o alinhamento do guia proposto às boas práticas de governança e gestão da segurança reconhecidas globalmente.

4.3.3 NIST Cybersecurity Framework (2024)

O NIST Cybersecurity Framework (CSF), desenvolvido pelo National Institute of Standards and Technology (NIST) dos Estados Unidos, é uma referência amplamente adotada para a gestão de riscos de segurança cibernética em organizações públicas e privadas. O framework organiza suas recomendações em cinco funções centrais: Identificar, Proteger, Detectar, Responder e Recuperar; que abrangem o ciclo completo de proteção e resposta a incidentes de segurança da informação. Sua adoção neste trabalho contribui para alinhar o guia proposto às melhores práticas internacionais de gestão de segurança, oferecendo uma base sólida para a integração entre requisitos normativos, técnicos e operacionais.

4.4 TRABALHOS ACADÊMICOS RELACIONADOS

Nesta seção será apresentado o trabalho acadêmico que dialoga com o tema deste estudo, oferecendo perspectivas teóricas e metodológicas já exploradas pela literatura. A análise dessa pesquisa permite compreender abordagens consolidadas,

identificar lacunas existentes e situar a contribuição deste trabalho no contexto científico.

4.4.1 Arquitetura integrada de normas e frameworks de desenvolvimento de software seguro aplicada para apoiar a identificação de falhas na especificação de requisitos (GATTO, 2024)

A tese de Gatto (2024) apresenta uma análise abrangente das principais normas e *frameworks* relacionados ao desenvolvimento de *software* seguro, identificando lacunas e sobreposições entre os trabalhos existentes. O autor propõe uma arquitetura integrada de normas e *frameworks* de desenvolvimento de *software* e segurança da informação, com o objetivo de aprimorar a identificação de falhas ainda na fase de especificação de requisitos, etapa fundamental do ciclo de vida de desenvolvimento seguro (S-SDLC). Embora o estudo tenha seu foco restrito à etapa de requisitos, ele fornece uma contribuição relevante ao propor uma base estruturada de controles fundamentada em três referências internacionais: ISO/IEC 27002:2022, NIST *Cybersecurity Framework* e CIS *Critical Security Controls*.

Essa abordagem complementa a proposta deste TCC ao evidenciar a importância de adaptar *frameworks* ao contexto específico do desenvolvimento de sistemas, buscando conciliar segurança, conformidade, aplicabilidade e clareza técnica para equipes de desenvolvimento.

4.5 INSTRUMENTO DE VERIFICAÇÃO

Os documentos apresentados nas seções anteriores constituem as fontes utilizadas na construção do guia proposto neste trabalho. Esta seção apresenta o documento selecionado como instrumento de verificação, o OWASP *Application Security Verification Standard* (ASVS), que, por desempenhar um papel metodológico distinto do das fontes de construção, não foi incluído nas etapas de análise comparativa e priorização descritas nas seções anteriores. A justificativa detalhada para essa escolha está na seção 2.5.

O ASVS é um documento publicado pela *Open Web Application Security Project* que define um conjunto de requisitos técnicos de segurança para o desenvolvimento e teste de aplicações *web* e móveis. Diferentemente dos demais

documentos da OWASP analisados neste trabalho, o ASVS parte da perspectiva da verificação: seu objetivo é definir o que uma aplicação precisa satisfazer para ser considerada segura em determinado nível de maturidade (OWASP, 2025a).

O documento organiza seus requisitos em três níveis progressivos de maturidade. O Nível 1 define os controles mínimos recomendados para qualquer aplicação, independentemente do contexto ou da criticidade dos dados tratados. O Nível 2 é destinado a aplicações que lidam com ataques menos comuns e que exigem controles mais rigorosos. O Nível 3 é reservado para aplicações de missão crítica, nas quais falhas de segurança podem ter consequências severas (OWASP, 2025a). Essa organização por níveis permite que o ASVS seja utilizado por organizações de diferentes graus de maturidade e contextos de uso, servindo tanto como referência de desenvolvimento quanto como definição de requisitos para aquisições de *software*.

Entre os casos de uso identificados pelo próprio documento, o ASVS pode ser utilizado como base para a elaboração de um guia de referência de codificação segura durante o desenvolvimento de aplicações, auxiliando desenvolvedores a manterem a segurança como preocupação constante ao longo do processo de construção do *software*. O próprio OWASP recomenda que, ao adotar o ASVS como base, organizações desenvolvam orientações próprias, claras e unificadas, idealmente elaboradas com apoio de profissionais de segurança (OWASP, 2025a). É precisamente esse caso de uso que fundamenta a relevância do ASVS para este trabalho: ao utilizar seus requisitos do Nível 1 como referência de verificação, verifica-se em que medida o guia proposto atende ao padrão mínimo esperado para um documento dessa natureza.

Neste trabalho, o ASVS não é utilizado como fonte de diretrizes para a construção do guia. Seu papel é restrito à etapa de verificação, descrita na seção 2.5, onde os requisitos do Nível 1 são utilizados como referência independente para verificar se as diretrizes consolidadas oferecem cobertura adequada frente a um padrão técnico reconhecido internacionalmente. Essa distinção de papel é o que garante que a avaliação não seja circular em relação ao processo de construção.

5 ANÁLISE E DISCUSSÃO

Este capítulo apresenta os resultados das três etapas da análise conduzida neste trabalho. Na primeira etapa, é apresentada a análise comparativa dos documentos selecionados na Revisão da Literatura, com o objetivo de identificar convergências, divergências e lacunas entre as diferentes fontes no que diz respeito às diretrizes de segurança aplicáveis ao desenvolvimento de aplicações governamentais. Na segunda etapa, as diretrizes identificadas são submetidas aos critérios de priorização definidos na seção 2.3, selecionando aquelas que atendem ao perfil técnico e de contexto estabelecido para o guia. Na terceira etapa, o conjunto de diretrizes resultante é confrontado com as dez categorias de vulnerabilidade do OWASP Top 10 (OWASP, 2025c), verificando se as ameaças mais comuns na prática de desenvolvimento estão sendo endereçadas. E, nos casos em que não estão, identificando se é possível complementar o guia dentro dos mesmos critérios de priorização já aplicados.

A análise comparativa foi conduzida com base na metodologia descrita na seção 2.2, utilizando as cinco categorias de classificação definidas e o critério de predominância funcional para alocação de cada diretriz. As diretrizes foram organizadas em subseções correspondentes a cada categoria, permitindo que convergências e lacunas sejam discutidas de forma contextualizada. A visão consolidada, com todas as categorias e fontes reunidas em uma única tabela, está disponível no Apêndice A deste trabalho.

É importante destacar que o objetivo da matriz comparativa não é hierarquizar as fontes nem atribuir julgamento sobre sua qualidade ou relevância, mas sim evidenciar quais aspectos de segurança cada documento prioriza, em que medida esses aspectos se alinham entre si e onde existem lacunas que o guia proposto neste trabalho poderá contribuir para preencher. A etapa de priorização não reavalia as fontes, mas aplica filtros de relevância técnica e contextual sobre o que foi identificado na etapa anterior. A verificação pelo OWASP Top 10 não é uma avaliação das fontes nem das diretrizes individualmente, mas sim do conjunto resultante das etapas anteriores, funcionando como um mecanismo de verificação orientado à ameaça, conforme descrito na seção 2.4.

5.1 GESTÃO DE IDENTIDADE E CONTROLE DE ACESSO

O Quadro 2 apresenta a comparação entre as fontes analisadas no que diz respeito à categoria de gestão de identidade e controle de acesso, que abrange diretrizes relacionadas à autenticação, autorização, gestão de sessões e aplicação do princípio do menor privilégio.

Quadro 2 – Matriz comparativa: Gestão de identidade e controle de acesso

Diretriz / Aspecto analisado	LGPD (2018)	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023)	Boas Práticas em Segurança da Informação (TCU, 2012)	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	OWASP Secure by Design (2025)	ISO/IEC 27001 (2022)	NIST Cybersecurity Framework (2024)	Tese de doutorado de Dacyr Dante de Oliveira Gatto (2024)
Autenticação e autorização de usuários	●	●	●	●	●	●	●	●
Princípio do menor privilégio	○	●	●	●	●	○	●	●
Gestão de sessões e tokens	○	●	●	●	●	○	○	●
Controle de acesso baseado em papéis (RBAC)	○	●	●	●	●	●	○	●

Nota: ● Aborda; ● Aborda parcialmente; ○ Não aborda

Fonte: elaborado pela autora

De modo geral, essa é uma das categorias com maior cobertura entre os documentos analisados, tanto na literatura técnica quanto nos marcos normativos, o que reflete o reconhecimento consolidado de que o controle sobre quem acessa o quê é um dos pilares fundamentais da segurança da informação.

A autenticação e autorização de usuários é o aspecto com convergência mais ampla nessa categoria: sete das oito fontes analisadas abordam esse aspecto de forma explícita. A LGPD (BRASIL, 2018) é a única fonte com cobertura parcial nesse aspecto, o que é esperado dado o seu caráter regulatório: ao estabelecer o princípio de segurança no artigo 46, ela cria um imperativo legal para que controles de autenticação sejam implementados, sem, contudo, prescrever mecanismos técnicos específicos.

O princípio do menor privilégio também apresenta convergência expressiva: o Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para

Aplicativos Móveis (PPSI, 2023), Boas Práticas em Segurança da Informação (TCU, 2012), Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024), OWASP Secure by Design (OWASP, 2025b) e Gatto (2024) o abordam explicitamente. O NIST CSF (NIST, 2024) trata esse aspecto de forma apenas parcial, ao referenciar o controle de acesso como resultado esperado sem detalhar como estruturar hierarquias de permissão mínimas no desenvolvimento de *software*. A ISO/IEC 27001 (ABNT, 2022) e a LGPD (BRASIL, 2018) não abordam esse aspecto, o que, no caso da norma ISO, é justificado pelo foco em requisitos de gestão organizacional, e não em práticas de codificação.

A gestão de sessões e *tokens* apresenta um padrão de cobertura mais heterogêneo entre as fontes. O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023), Boas Práticas em Segurança da Informação (TCU, 2012) e Gatto (2024) tratam esse aspecto explicitamente, demonstrando que tanto os documentos nacionais mais especializados quanto a literatura acadêmica reconhecem a importância do gerenciamento de sessão como controle técnico fundamental. O Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024) e o OWASP Secure by Design (OWASP, 2025b) abordam o tema de forma parcial, enquanto ISO/IEC 27001 (ABNT, 2022), NIST CSF (NIST, 2024) e LGPD (BRASIL, 2018) não o contemplam. A ausência desse aspecto nas normas de gestão reflete a diferença de escopo já mencionada: normas orientadas a resultados organizacionais tendem a não detalhar controles técnicos de sessão, que são mais próprios de guias de implementação.

O controle de acesso baseado em papéis (RBAC) é, dos quatro aspectos dessa categoria, aquele com maior variação na cobertura. Gatto (2024) e o Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024) são as únicas fontes com cobertura explícita. Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023), Boas Práticas em Segurança da Informação (TCU, 2012), OWASP Secure by Design (OWASP, 2025b) e ISO/IEC 27001 (ABNT, 2022) abordam o RBAC de forma parcial, enquanto NIST CSF (NIST, 2024) e LGPD (BRASIL, 2018) não o contemplam. Essa distribuição sugere que o RBAC, embora seja um padrão consolidado de controle de acesso, ainda é tratado de forma incompleta pela maioria das fontes normativas e regulatórias analisadas, o que reforça a necessidade de que o guia proposto inclua orientações

concretas sobre como estruturar papéis e permissões no contexto de aplicações governamentais.

5.2 PROTEÇÃO DE DADOS E PRIVACIDADE

O Quadro 3 apresenta a comparação entre as fontes analisadas no que diz respeito à categoria de proteção de dados e privacidade, que compreende diretrizes voltadas à criptografia, minimização de dados, consentimento do titular, privacidade por *design* e armazenamento seguro de credenciais.

Quadro 3 – Matriz comparativa: Proteção de dados e privacidade

Diretriz / Aspecto analisado	LGPD (2018)	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023)	Boas Práticas em Segurança da Informação (TCU, 2012)	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	OWASP Secure by Design (2025)	ISO/IEC 27001 (2022)	NIST Cybersecurity Framework (2024)	Tese de doutorado de Dacyr Dante de Oliveira Gatto (2024)
Criptografia de dados em repouso e em trânsito	●	●	●	●	●	●	○	●
Minimização e anonimização de dados	●	●	○	●	●	○	○	●
Consentimento e direitos do titular (LGPD)	●	●	○	●	○	○	○	○
Privacy by design e privacy by default	●	●	○	●	●	○	○	○
Armazenamento seguro de credenciais e segredos	○	●	●	●	●	●	●	●

Nota: ● Aborda; ● Aborda parcialmente; ○ Não aborda

Fonte: elaborado pela autora

Essa é a categoria em que a LGPD (BRASIL, 2018) apresenta sua cobertura mais expressiva. Os aspectos de consentimento e direitos do titular e de minimização e anonimização de dados são abordados explicitamente pela Lei, especialmente em seus artigos 6º, 7º e 46º, que estabelecem os princípios de finalidade, necessidade, adequação e segurança como requisitos vinculantes para qualquer sistema que trate dados pessoais. A privacidade por design recebe cobertura parcial na LGPD, por estar implícita no artigo 46, mas sem o detalhamento técnico de sua implementação. Nesse sentido, a LGPD funciona nessa categoria menos como um guia de implementação e mais como um conjunto de obrigações legais que os demais documentos ajudam a operacionalizar tecnicamente.

O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) e o Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024) são as fontes com cobertura mais ampla nessa categoria, sendo as únicas que abordam explicitamente a criptografia de dados em repouso e em trânsito, a minimização de dados, o consentimento e a privacidade por *design* simultaneamente. O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) é especialmente relevante por ter sido elaborado especificamente para o contexto de aplicativos governamentais e por já incorporar os princípios da LGPD em recomendações técnicas concretas. Sua orientação sobre privacidade por concepção evidencia que mesmo decisões arquitetônicas aparentemente pequenas podem ter grande influência na privacidade e na proteção de dados pessoais (PPSI, 2023), reforçando a necessidade de tratar a privacidade como requisito de projeto. O Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024), por sua vez, amplia essa perspectiva ao trazer orientações para o contexto mais amplo da Administração Pública Federal.

A ISO/IEC 27001 (ABNT, 2022) e o documento de Boas Práticas em Segurança da Informação (TCU, 2012) apresentam o mesmo perfil nessa categoria: cobertura parcial para criptografia de dados e para armazenamento seguro de credenciais, e ausência de cobertura para minimização de dados, consentimento do titular e privacidade por *design*. Esse alinhamento entre uma norma de gestão internacional e um documento nacional de orientação operacional sugere que, embora produzidos em contextos distintos, ambos tratam a proteção de dados com foco em controles técnicos de criptografia e armazenamento, sem incorporar os princípios de privacidade por concepção que passaram a ser exigidos pela LGPD. No caso do TCU (2012), a ausência dos aspectos relacionados à LGPD é diretamente explicada pelo período de publicação, anterior à Lei. No caso da ISO/IEC 27001, a ausência reflete o caráter internacional do documento, que não foi concebido para o contexto regulatório brasileiro e tende a tratar privacidade como um domínio separado de gestão, e não como requisito técnico integrado ao desenvolvimento de *software*.

O OWASP *Secure by Design Framework* (OWASP, 2025b) aborda todos os itens de maneira parcial, mas não contempla consentimento nem direitos do titular previstos na LGPD. Esse comportamento é coerente com o foco do documento nas

práticas de engenharia de *software* seguro, sem pretensão de cobrir obrigações legais específicas de jurisdições nacionais.

O NIST CSF (NIST, 2024) apresenta o perfil mais restritivo dessa categoria entre as fontes analisadas: é o único documento que não aborda sequer a criptografia de dados de forma explícita ou parcial, registrando cobertura nula nesse aspecto, e mantém cobertura parcial apenas para armazenamento seguro de credenciais. Os demais aspectos também não são contemplados. Essa ausência é coerente com a natureza do NIST CSF como *framework* orientado a resultados de gestão de risco cibernético, cujas funções (Identificar, Proteger, Detectar, Responder e Recuperar) descrevem o que deve ser alcançado sem prescrever controles técnicos específicos de implementação. Para o guia proposto, isso significa que o NIST CSF oferece pouca base direta para as diretrizes dessa categoria, sendo necessário complementá-lo com as fontes que detalham a implementação técnica dos controles de proteção de dados.

Gatto (2024) aborda explicitamente criptografia e armazenamento seguro de credenciais, contempla apenas parcialmente a minimização de dados, mas não aborda consentimento nem privacidade por *design*. Esse perfil é consistente com o foco da tese em aspectos técnicos do ciclo de desenvolvimento, sem abordar obrigações legais ou princípios de privacidade por concepção.

5.3 VALIDAÇÃO DE ENTRADAS E SEGURANÇA NA IMPLEMENTAÇÃO

O Quadro 4 apresenta a comparação entre as fontes analisadas no que diz respeito à categoria de validação de entradas e segurança na implementação, que abrange práticas de codificação segura, prevenção de vulnerabilidades comuns, tratamento de erros e uso de protocolos seguros de comunicação.

Quadro 4 – Matriz comparativa: Validação de entradas e segurança na implementação

Diretriz / Aspecto analisado	LGPD (2018)	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023)	Boas Práticas em Segurança da Informação (TCU, 2012)	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	OWASP Secure by Design (2025)	ISO/IEC 27001 (2022)	NIST Cybersecurity Framework (2024)	Tese de doutorado de Dacyr Dante de Oliveira Gatto (2024)
Validação e sanitização de entradas	○	●	●	●	○	○	○	●
Codificação segura e padrões OWASP	○	●	●	●	○	●	●	●
Tratamento seguro de erros e logs	○	●	●	●	●	●	●	●
Comunicação segura (TLS/HTTPS)	○	●	○	●	●	●	●	●

Nota: ● Aborda; ● Aborda parcialmente; ○ Não aborda
Fonte: elaborado pela autora

Essa categoria concentra as diretrizes mais diretamente voltadas à atividade cotidiana do desenvolvedor de *software*. Gatto (2024) é a fonte com cobertura mais completa nessa categoria, abordando de forma explícita todos os aspectos. O Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024) apresenta cobertura explícita para codificação segura, tratamento de erros e *logs* e comunicação segura via TLS, e cobertura parcial para validação de entradas, configurando-se como uma referência técnica relevante para o contexto governamental.

A LGPD (BRASIL, 2018) não apresenta cobertura sobre nenhum dos aspectos técnicos dessa categoria, o que é esperado dado o seu caráter regulatório. Sua relevância para essa categoria é indireta: ao estabelecer obrigações de segurança no tratamento de dados pessoais (art. 46), ela cria um imperativo legal para que as práticas de codificação segura, validação de entradas e tratamento correto de erros sejam efetivamente implementadas.

O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) aborda explicitamente codificação segura e comunicação segura via TLS, aspectos que refletem seu foco em orientações práticas para o desenvolvimento de aplicativos móveis. Essa distribuição é coerente com o perfil do documento, voltado a orientações diretas ao desenvolvedor de aplicativos, sem a profundidade técnica de *frameworks* específicos de desenvolvimento seguro.

O documento de Boas Práticas em Segurança da Informação (TCU, 2012) apresenta cobertura parcial para validação de entradas, codificação segura e tratamento de erros, mas não aborda comunicação segura. A ausência de TLS é uma lacuna relevante, especialmente considerando que essa é uma prática considerada requisito mínimo em aplicações que trafegam dados sensíveis. Essa ausência, contudo, pode ser parcialmente explicada pelo período de publicação do documento, quando o uso generalizado de HTTPS ainda não era uma exigência tão consolidada quanto é hoje.

O OWASP *Secure by Design Framework* (OWASP, 2025b) apresenta uma das coberturas mais restritas nessa categoria, considerando que se trata de um documento voltado especificamente ao desenvolvimento seguro de *software*. Esse comportamento pode ser explicado pelo foco do documento: o OWASP *Secure by Design* opera predominantemente em nível de princípios arquiteturais e de *design*, como redução da superfície de ataque e segurança por padrão, sem descer ao nível de práticas de implementação linha a linha. Nesse sentido, o documento complementa, mas não substitui, referências mais orientadas à codificação, como o OWASP *Secure Coding Practices*, que detalha controles técnicos específicos para validação de entradas e prevenção de vulnerabilidades de implementação.

A ISO/IEC 27001 (ABNT, 2022) apresenta cobertura explícita para tratamento de erros e *logs* (alinhada à sua ênfase em controles de auditoria), e cobertura parcial para codificação segura e comunicação via TLS (aspectos que a norma reconhece como controles relevantes, mas trata em nível de requisito organizacional, sem detalhar práticas de implementação). Validação e sanitização de entradas não são abordadas porque a ISO/IEC 27001 não opera no nível das práticas de codificação linha a linha, mas sim no nível de gestão de riscos e conformidade, diferentemente de outras referências técnicas.

O NIST CSF (NIST, 2024) apresenta cobertura explícita para comunicação segura e cobertura parcial para codificação segura e tratamento de erros. Assim como a ISO, o NIST opera em nível de resultado esperado, sem detalhar práticas específicas de implementação, o que limita sua utilidade direta para equipes de desenvolvimento.

5.4 GESTÃO DE VULNERABILIDADES E ATUALIZAÇÕES

O Quadro 5 apresenta a comparação entre as fontes analisadas no que diz respeito à categoria de gestão de vulnerabilidades e atualizações, que inclui diretrizes voltadas à identificação e correção de falhas, gestão de dependências de terceiros, resposta a incidentes, monitoramento contínuo, testes de segurança e continuidade e recuperação de dados.

Quadro 5 – Matriz comparativa: Gestão de vulnerabilidades e atualizações

Diretriz / Aspecto analisado	LGPD (2018)	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação (BRASIL, 2020)	Boas Práticas em Segurança da Informação (TCU, 2012)	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	OWASP Secure by Design (2025)	ISO/IEC 27001 (2022)	NIST Cybersecurity Framework (2024)	Tese de doutorado de Dacyr Dante de Oliveira Gatto (2024)
Identificação e correção de vulnerabilidades	○	○	●	●	●	●	●	●
Gestão de dependências e componentes de terceiros (SCA)	○	●	○	●	●	●	●	●
Resposta a incidentes de segurança	●	●	●	●	●	●	●	●
Monitoramento contínuo e alertas	●	○	●	●	●	●	●	●
Testes de segurança (SAST, DAST, pentest)	○	●	○	●	●	●	●	●
Continuidade e recuperação de dados (backup, PCN)	●	●	●	●	○	●	●	●

Nota: ● Aborda; ● Aborda parcialmente; ○ Não aborda

Fonte: elaborado pela autora

Essa é a categoria em que a ISO/IEC 27001 (ABNT, 2022) e o NIST CSF (NIST, 2024) apresentam cobertura mais expressiva, abordando explicitamente identificação e correção de vulnerabilidades, gestão de dependências, resposta a incidentes, monitoramento contínuo, testes de segurança e continuidade de dados. Essa abrangência reflete o foco dessas normas em gestão de riscos e continuidade operacional. Gatto (2024) e o Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024) também apresentam cobertura explícita em quase todos os aspectos da categoria.

A LGPD (BRASIL, 2018) apresenta cobertura parcial para resposta a incidentes e monitoramento, ao estabelecer em seu artigo 48 a obrigação de

comunicar incidentes de segurança à ANPD e aos titulares quando houver risco ou dano relevante, e cobertura parcial para continuidade de dados. Esse requisito legal precisa ser contemplado nos planos de resposta das equipes de desenvolvimento, reforçando a necessidade de articular conformidade legal e práticas técnicas de forma integrada.

O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) apresenta cobertura parcial para gestão de dependências, resposta a incidentes, testes de segurança e continuidade de dados, mas não aborda identificação e correção de vulnerabilidades nem monitoramento contínuo. Essa lacuna é relevante para o escopo deste trabalho, dado que aplicações governamentais demandam visibilidade contínua sobre seu estado de segurança ao longo de todo o ciclo de vida, não apenas durante o desenvolvimento inicial.

O documento de Boas Práticas em Segurança da Informação (TCU, 2012) apresenta cobertura explícita para resposta a incidentes e continuidade e recuperação de dados, e cobertura parcial para identificação de vulnerabilidades e monitoramento. A ausência de cobertura sobre gestão de dependências de componentes de terceiros e sobre testes de segurança é uma limitação relevante, especialmente considerando que aplicações governamentais frequentemente incorporam bibliotecas externas. Essa ausência pode ser atribuída ao período de publicação do documento, anterior à popularização de ecossistemas de pacotes gerenciados e às práticas modernas de *Software Composition Analysis* (SCA).

O OWASP *Secure by Design* (OWASP, 2025b) aborda identificação de vulnerabilidades, gestão de dependências, resposta a incidentes, testes e monitoramento de forma parcial, refletindo seu escopo centrado nas fases de design e implementação. A ausência de cobertura explícita sobre continuidade e recuperação de dados é coerente com o foco do documento em práticas de construção de *software*, sem pretensão de cobrir aspectos de infraestrutura operacional.

5.5 ARQUITETURA E DESIGN SEGURO

O Quadro 6 apresenta a comparação entre as fontes analisadas no que diz respeito à categoria de arquitetura e *design* seguro, que contempla diretrizes relacionadas à modelagem de ameaças, redução da superfície de ataque, segurança por padrão, segmentação de componentes e rastreabilidade de ações.

Quadro 6 – Matriz comparativa: Arquitetura e design seguro

Diretriz / Aspecto analisado	LGPD (2018)	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023)	Boas Práticas em Segurança da Informação (TCU, 2012)	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	OWASP Secure by Design (2025)	ISO/IEC 27001 (2022)	NIST Cybersecurity Framework (2024)	Tese de doutorado de Dacyr Dante de Oliveira Gatto (2024)
Modelagem de ameaças (Threat Modeling)	○	○	○	●	●	●	●	●
Redução da superfície de ataque	○	●	○	●	●	●	●	●
Segurança por padrão (Secure by Default)	●	●	○	●	●	○	○	●
Segmentação e isolamento de componentes	○	●	○	●	●	●	○	○
Rastreabilidade e auditoria de ações	●	●	●	●	●	●	●	●

Nota: ● Aborda; ● Aborda parcialmente; ○ Não aborda

Fonte: elaborado pela autora

Essa categoria reúne as diretrizes que atuam mais cedo no SDLC, influenciando decisões tomadas nas fases de especificação e projeto antes que qualquer linha de código seja escrita. É também a categoria em que se observam algumas das lacunas mais significativas entre os documentos analisados.

A rastreabilidade e auditoria de ações é o aspecto com maior convergência nessa categoria: Boas Práticas em Segurança da Informação (TCU, 2012), Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024), OWASP *Secure by Design* (OWASP, 2025b), ISO/IEC 27001 (ABNT, 2022) e NIST CSF (NIST, 2024) abordam esse aspecto de forma explícita, enquanto LGPD (BRASIL, 2018), Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) e Gatto (2024) o tratam de forma parcial. No caso do TCU (2012), a ênfase em rastreabilidade reforça a publicação como referência de controle e responsabilização no uso de sistemas pela Administração Pública. A LGPD, por sua vez, impõe por meio do artigo 37 a obrigação de manutenção de registros das operações de tratamento de dados pessoais, o que, embora não seja uma diretriz técnica de arquitetura, influencia diretamente o projeto de sistemas no que diz respeito à implementação de *logs* de auditoria.

O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) não contempla modelagem de ameaças e

aborda os outros itens de forma parcial. Essa lacuna pode ser atribuída ao foco mais operacional do documento, voltado a orientações diretas ao desenvolvedor, sem aprofundar aspectos de arquitetura que exigem decisões em nível de projeto de sistema.

O documento de Boas Práticas em Segurança da Informação (TCU, 2012) apresenta cobertura explícita apenas para rastreabilidade, não abordando os outros itens. Isso indica que o documento não foi concebido para orientar decisões técnicas de arquitetura de *software*, sendo mais adequado como referência de controles operacionais e de gestão de segurança.

O OWASP Secure by Design (OWASP, 2025b) e o Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024) são as fontes com cobertura mais expressiva nessa categoria, abordando explicitamente redução da superfície de ataque, segurança por padrão, segmentação de componentes e rastreabilidade. O OWASP fundamenta sua abordagem nos princípios de *Security by Design*, defendendo que esses quatro aspectos devem ser requisitos arquiteturais não negociáveis desde as fases iniciais de concepção do sistema. O Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024) contextualiza esses princípios no ambiente da Administração Pública Federal, onde requisitos de transparência, responsabilização e auditabilidade impõem exigências adicionais sobre o projeto dos sistemas.

A ISO/IEC 27001 (ABNT, 2022) aborda rastreabilidade de forma explícita e os outros itens de forma parcial ou não aborda, assim como o NIST CSF (NIST, 2024). Em ambos os casos, a modelagem de ameaças é referenciada principalmente no contexto de gestão de riscos organizacionais, e não como prática técnica de projeto de *software*, o que limita seu uso como guia de implementação.

Gatto (2024) apresenta cobertura explícita para redução da superfície de ataque, e parcial para modelagem de ameaças, e segurança por padrão, mas não aborda segmentação de componentes. A tese demonstra, empiricamente, que a ausência de modelagem de ameaças na fase de requisitos é um dos principais vetores de introdução de falhas de segurança em sistemas de *software* (GATTO, 2024), argumento que reforça a centralidade dessa prática no guia proposto.

De forma geral, a análise dessa categoria evidencia que a modelagem de ameaças é o aspecto com menor cobertura entre todos os documentos: nenhuma fonte a aborda de forma explícita como prática técnica de projeto de *software*, sendo

tratada apenas parcialmente na maioria dos documentos. Essa lacuna generalizada reforça a pertinência de incluir a modelagem de ameaças como diretriz central no guia proposto.

5.6 SÍNTESE DA ANÁLISE COMPARATIVA

Como demonstra os itens acima, a análise comparativa conduzida neste capítulo evidencia que nenhuma das fontes analisadas apresenta cobertura completa em todas as categorias e aspectos avaliados. Cada documento reflete, em sua estrutura e em suas lacunas, o contexto em que foi produzido: normas de gestão oferecem ampla cobertura em gestão de vulnerabilidades e rastreabilidade, mas não detalham práticas de implementação técnica; guias orientados ao desenvolvimento, como o OWASP *Secure by Design Framework* (OWASP, 2025b) e a tese de Gatto (2024), aprofundam aspectos de arquitetura e codificação, mas apresentam lacunas em proteção de dados e conformidade regulatória; documentos nacionais trazem aderência ao contexto governamental brasileiro, mas não cobrem integralmente as exigências técnicas contemporâneas; e a própria LGPD (BRASIL, 2018), embora seja a referência normativa central para o tratamento de dados pessoais no país, não prescreve controles técnicos de implementação.

Essa heterogeneidade não é uma falha das fontes individualmente, mas uma característica inerente à diversidade de seus propósitos, escopos e períodos de publicação. O que ela revela, no entanto, é que nenhum documento isolado é capaz de orientar de forma suficiente o desenvolvimento seguro de aplicações governamentais que sejam simultaneamente técnicos, conformes à legislação nacional e adequados ao perfil dos usuários cidadãos. É justamente essa lacuna que justifica a proposta deste trabalho: a construção de um guia que articule, em um conjunto coeso de diretrizes priorizadas, as contribuições complementares das diferentes fontes analisadas, preenchendo as ausências de cada uma com o que as demais têm a oferecer.

6 DESENVOLVIMENTO DO GUIA

Esta seção apresenta o processo de construção do guia de diretrizes de segurança da informação proposto neste trabalho. O objetivo do guia é oferecer um conjunto organizado de recomendações práticas voltado a desenvolvedores de aplicações governamentais, consolidando as principais diretrizes identificadas ao longo da pesquisa de forma acessível e aplicável a desenvolvedores de *software*.

As diretrizes apresentadas no guia estão organizadas com base nas fases do SDLC (requisitos, projeto, codificação, testes e implantação e manutenção) por se tratar de uma estrutura conceitual amplamente reconhecida que descreve as atividades essenciais presentes em qualquer processo de desenvolvimento de *software*, independente da metodologia adotada. Equipes que trabalham com abordagens iterativas, como métodos ágeis, podem aplicar as diretrizes de cada fase no momento do ciclo em que aquela atividade ocorre, revisitando-as a cada iteração conforme necessário. Equipes que adotam práticas de DevSecOps encontrarão nas diretrizes de codificação e testes orientações compatíveis com a integração contínua de segurança ao *pipeline* de desenvolvimento.

O objetivo é que o guia seja útil independentemente do processo adotado, sem exigir mudanças na forma como a equipe já trabalha e sem necessitar da leitura deste trabalho de conclusão de curso. As seções a seguir descrevem as decisões tomadas para sua elaboração, os critérios aplicados na seleção das diretrizes e as contribuições de cada fonte utilizada como referência.

6.1 DECISÕES DE DESENVOLVIMENTO

A construção do guia partiu de duas decisões centrais. A primeira diz respeito ao formato do artefato: optou-se por um guia de diretrizes e boas práticas, e não por um *framework*, tendo em vista que o objetivo do trabalho é oferecer orientações claras e acessíveis para desenvolvedores, sem impor uma estrutura metodológica rígida que exigiria adaptações organizacionais mais amplas, como já discutido na seção 3.1. Essa escolha permite que as recomendações sejam adotadas de forma flexível, independentemente do modelo de desenvolvimento utilizado pela equipe.

A segunda decisão diz respeito ao escopo das diretrizes: foram incluídas apenas recomendações com aplicabilidade direta ao desenvolvimento de *software*,

excluindo práticas de caráter exclusivamente gerencial ou institucional que não se traduzem em ações concretas para equipes de desenvolvimento. Essa delimitação está alinhada ao critério de aplicabilidade prática e aderência ao SDLC, definido na seção 2.3.

6.2 APLICAÇÃO DOS CRITÉRIOS DE PRIORIZAÇÃO

A seleção das diretrizes que compõem o guia foi realizada com base nos quatro critérios de priorização definidos na seção 2.3: impacto na segurança, aplicabilidade prática e aderência ao SDLC, relevância normativa e recorrência nas fontes, e adequação ao contexto de aplicações governamentais. Foram incluídas no guia as diretrizes que atenderam a pelo menos dois desses critérios e que apresentam aplicabilidade concreta ao ciclo de desenvolvimento de *software*, garantindo que as recomendações sejam tecnicamente relevantes, acionáveis pela equipe de desenvolvimento e adequadas ao escopo do trabalho.

Cabe destacar que nem todos os aspectos analisados na matriz comparativa foram convertidos em diretrizes do guia. A matriz comparativa, apresentada no Capítulo 5, tem como objetivo mapear a cobertura das fontes sobre cada aspecto de segurança, incluindo temas recorrentes independentemente de sua tradução direta em ações de desenvolvimento. Já a matriz de priorização aplica um filtro adicional orientado ao escopo definido na seção 6.1: são selecionadas apenas as diretrizes que o desenvolvedor pode implementar de forma concreta ao longo do SDLC. Aspectos de natureza predominantemente gerencial ou institucional (como políticas organizacionais de segurança, gestão de continuidade do negócio e classificação formal de informações) foram identificados na análise comparativa e registrados no quadro abaixo, mas não foram incluídos no guia por extrapolarem o escopo de atuação de equipes de desenvolvimento. Isso não significa que tais aspectos sejam menos relevantes, mas que sua implementação depende de decisões organizacionais que vão além do que um guia voltado ao desenvolvedor pode endereçar.

O Quadro 7 apresenta o resultado dessa avaliação para todas as diretrizes identificadas na análise comparativa, indicando quais critérios cada uma atende, sua situação de inclusão e a respectiva justificativa. Essa visão consolidada permite rastrear a fundamentação de cada decisão e evidencia que a composição do guia não

foi arbitrária, mas apoiada em critérios objetivos e consistentes com a metodologia adotada.

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamental	Total	Situação	Justificativa
Categoria 1 — Gestão de identidade e controle de acesso							
Autenticação e autorização de usuários	✓	✓	✓	✓	4/4	Incluída	C1: falhas geram acesso não autorizado a dados e sistemas. C2: implementável nas fases de requisitos e codificação. C3: presente em normas, legislação e frameworks reconhecidos. C4: aplicações governamentais expõem dados de cidadãos, exigindo controle rigoroso de acesso.
Princípio do menor privilégio	✓	✓	✓	✓	4/4	Incluída	C1: permissões excessivas ampliam a superfície de ataque e o risco de comprometimento. C2: decisão de design implementável na fase de projeto e codificação. C3: presente em legislação, normas e frameworks reconhecidos. C4: órgãos públicos operam com dados sensíveis de cidadãos, tornando a restrição de privilégios uma medida de responsabilização.
Gestão de sessões e tokens	✓	✓	✓	—	3/4	Incluída	C1: sessões mal gerenciadas favorecem sequestro de contas e acesso indevido. C2: ação direta na fase de codificação. C3: presente em normas e guias técnicos reconhecidos. C4: não aplicável, pois não há obrigação ou risco específico do setor público que a diferencie de qualquer outro contexto.
Controle de acesso baseado em papéis (RBAC)	✓	✓	✓	✓	4/4	Incluída	C1: ausência de hierarquia de permissões aumenta o risco de acesso indevido a dados e funcionalidades. C2: implementável na fase de projeto e codificação. C3: presente em normas e frameworks reconhecidos. C4: sistemas governamentais frequentemente têm múltiplos perfis de usuário com diferentes níveis de acesso a dados públicos e sigilosos.

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamental	Total	Situação	Justificativa
Categoria 2 — Proteção de dados e privacidade							
Criptografia de dados em repouso e em trânsito	✓	✓	✓	✓	4/4	Incluída	C1: ausência expõe dados sensíveis a interceptação e acesso não autorizado. C2: implementável nas fases de projeto e codificação. C3: presente em legislação, normas e frameworks reconhecidos. C4: aplicações governamentais tratam dados pessoais de cidadãos, sujeitos à LGPD.
Minimização e anonimização de dados	✓	✓	✓	✓	4/4	Incluída	C1: reduz a superfície de exposição em caso de incidente. C2: decisão implementável na fase de requisitos e design. C3: obrigação legal expressa na LGPD. C4: sistemas do governo estão diretamente sujeitos à LGPD e às obrigações do PPSI.
Consentimento e direitos do titular (LGPD)	—	✓	✓	✓	3/4	Incluída	C1: não aplicável, pois sua ausência não gera vulnerabilidade técnica direta, mas descumprimento legal. C2: mecanismos de consentimento são implementáveis na fase de codificação. C3: obrigação legal expressa na LGPD. C4: aplicações governamentais tratam dados pessoais de cidadãos sob regime de cumprimento obrigatório da LGPD.

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamental	Total	Situação	Justificativa
Privacy by design e privacy by default	✓	✓	✓	✓	4/4	Incluída	C1: decisões arquiteturais inadequadas aumentam o risco de exposição de dados pessoais. C2: princípio aplicável desde a fase de requisitos e projeto. C3: previsto na LGPD e em frameworks reconhecidos. C4: devido à sensibilidade de dados e aos ataques a que sistemas governamentais estão sujeitos, deve-se incorporar privacidade desde a concepção.
Armazenamento seguro de credenciais e segredos	✓	✓	✓	✓	4/4	Incluída	C1: credenciais expostas comprometem a confidencialidade e viabilizam ataques de larga escala. C2: ação direta na fase de codificação. C3: presente em normas e frameworks reconhecidos. C4: aplicações governamentais concentram credenciais de grande volume de usuários cidadãos.
Categoria 3 — Validação de entradas e segurança na implementação							
Validação e sanitização de entradas	✓	✓	✓	—	3/4	Incluída	C1: ausência favorece injeção, XSS e outras vulnerabilidades críticas. C2: ação direta na fase de codificação. C3: presente em guias técnicos e frameworks reconhecidos. C4: não aplicável, pois é uma prática universal de codificação sem especificidade ao contexto governamental.
Codificação segura e padrões OWASP	✓	✓	✓	✓	4/4	Incluída	C1: práticas inseguras introduzem vulnerabilidades exploráveis. C2: ação central na fase de codificação. C3: presente em normas, guias técnicos e frameworks reconhecidos. C4: aplicações governamentais são alvos frequentes por concentrarem dados sensíveis de cidadãos.

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamenta	Total	Situação	Justificativa
Tratamento seguro de erros e logs	✓	✓	✓	✓	4/4	Incluída	C1: erros mal tratados expõem informações do sistema; logs inadequados comprometem a rastreabilidade. C2: ação direta nas fases de codificação e implantação. C3: presente em normas e guias técnicos reconhecidos. C4: a LGPD impõe obrigação de registro das operações de tratamento de dados.
Comunicação segura (TLS/HTTPS)	✓	✓	✓	✓	4/4	Incluída	C1: ausência expõe dados em trânsito a interceptação. C2: configuração implementável nas fases de codificação e implantação. C3: presente em legislação, normas e frameworks reconhecidos. C4: aplicações governamentais trafegam dados pessoais sensíveis de cidadãos, tornando TLS um requisito mínimo.
Categoria 4 — Gestão de vulnerabilidades e atualizações							
Identificação e correção de vulnerabilidades	✓	✓	✓	✓	4/4	Incluída	C1: falhas não corrigidas são vetores diretos de comprometimento. C2: atividade da fase de manutenção do SDLC. C3: presente em normas e frameworks reconhecidos. C4: sistemas governamentais são alvos frequentes, exigindo resposta ágil a vulnerabilidades.
Gestão de dependências e componentes de terceiros (SCA)	✓	✓	✓	—	3/4	Incluída	C1: componentes desatualizados introduzem vulnerabilidades conhecidas. C2: ação na fase de codificação e manutenção. C3: presente em normas e guias técnicos reconhecidos. C4: não aplicável, pois o risco de dependências não é específico ao setor público.

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamental	Total	Situação	Justificativa
Resposta a incidentes de segurança	✓	✓	✓	✓	4/4	Incluída	C1: resposta inadequada amplia o impacto de um incidente sobre confidencialidade e disponibilidade. C2: a dimensão técnica da resposta é implementável nas fases de codificação e implantação. C3: obrigação legal expressa na LGPD. C4: a notificação à ANPD e aos titulares é obrigação específica do tratamento de dados pessoais por órgãos públicos.
Monitoramento contínuo e alertas	✓	✓	✓	✓	4/4	Incluída	C1: ausência impede detecção de ataques e comportamentos anômalos. C2: configuração de logs estruturados e alertas é implementável pelo desenvolvedor nas fases de implantação e manutenção. C3: presente em normas e frameworks reconhecidos. C4: aplicações governamentais, por processarem dados pessoais de cidadãos em larga escala, têm obrigação reforçada de manter visibilidade sobre o comportamento do sistema.
Testes de segurança (SAST, DAST, pentest)	✓	✓	✓	—	3/4	Incluída	C1: ausência de testes permite que vulnerabilidades cheguem à produção. C2: atividade central da fase de testes do SDLC. C3: presente em normas e frameworks reconhecidos. C4: não aplicável, pois testes de segurança são necessários em qualquer contexto de desenvolvimento, sem especificidade governamental.

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamental	Total	Situação	Justificativa
Continuidade e recuperação de dados (backup, PCN)	✓	—	✓	✓	3/4	Não incluída	C1: ausência compromete a disponibilidade dos serviços. C2: não aplicável, pois envolve estratégias de backup e recuperação de desastres que dependem de infraestrutura e operações, fora do escopo de ação da equipe de desenvolvimento. C3: presente em normas reconhecidas. C4: serviços governamentais têm obrigação de disponibilidade contínua para o cidadão.
Categoria 5 — Arquitetura e design seguro							
Modelagem de ameaças (Threat Modeling)	✓	✓	✓	✓	4/4	Incluída	C1: sua ausência na fase de projeto é vetor direto de introdução de falhas de segurança. C2: prática aplicável desde a fase de requisitos e projeto. C3: presente parcialmente em normas e frameworks reconhecidos. C4: sistemas governamentais processam dados pessoais sensíveis de cidadãos, impondo exigências adicionais sobre o projeto.
Redução da superfície de ataque	✓	✓	✓	✓	4/4	Incluída	C1: minimiza os vetores disponíveis para exploração. C2: decisão aplicável nas fases de requisitos e projeto. C3: presente em frameworks reconhecidos. C4: aplicações governamentais são alvos de alto valor por concentrarem dados e serviços de interesse público.
Segurança por padrão (Secure by Default)	✓	✓	✓	✓	4/4	Incluída	C1: configurações inseguras por padrão aumentam o risco de exposição. C2: decisão arquitetural aplicável nas fases de projeto e implantação. C3: presente em frameworks reconhecidos. C4: sistemas governamentais devem operar com configurações seguras desde a implantação para proteger dados de cidadãos.

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

Quadro 7 – Matriz de priorização de diretrizes

(continua)

Aspecto analisado	C1 Impacto	C2 SDLC	C3 Normativa	C4 Governamental	Total	Situação	Justificativa
Segmentação e isolamento de componentes	✓	✓	✓	✓	4/4	Incluída	C1: falha em um componente isolado não compromete o sistema inteiro. C2: decisão de arquitetura aplicável na fase de projeto. C3: presente em normas e frameworks reconhecidos. C4: sistemas governamentais com múltiplos módulos e perfis de acesso se beneficiam diretamente do isolamento de responsabilidades.
Rastreabilidade e auditoria de ações	✓	✓	✓	✓	4/4	Incluída	C1: ausência de logs de auditoria compromete a rastreabilidade e a capacidade de resposta a incidentes. C2: implementável nas fases de codificação e implantação. C3: obrigação legal expressa na LGPD. C4: a Administração Pública Federal tem exigência institucional de transparência e responsabilização no uso de sistemas.

Fonte: elaborado pela autora

Comentado [1]: palavras ficaram quebradas. Será que não é melhor essa tabela ser inserida na horizontal nas próximas páginas?

Comentado [2]: @luizbohlke@hotmail.com por favor confirma que eu posso inserir uma página com uma orientação (paisagem) diferente. ACHO que seguindo as margens e medidas que as normas exigem, não tem problema

6.3 BASE DE REFERÊNCIA DAS DIRETRIZES

As diretrizes consolidadas no guia foram extraídas das fontes analisadas na seção 5, Análise e Discussão, onde as convergências, divergências e lacunas entre os documentos foram detalhadas por categoria. A seguir, é apresentada uma síntese da contribuição de cada fonte para a composição do guia, retomando os achados discutidos no capítulo anterior.

A LGPD (BRASIL, 2018) forneceu a base normativa para as diretrizes relacionadas à proteção de dados pessoais, consentimento, minimização de dados e transparência no tratamento das informações. Por tratar-se de legislação de cumprimento obrigatório, suas exigências foram consideradas de alta relevância normativa e de adequação ao contexto governamental, ainda que sua abordagem seja predominantemente regulatória e não técnica.

O Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023) contribuiu de forma direta com diretrizes orientadas ao contexto específico de aplicativos governamentais móveis, especialmente nas categorias de gestão de identidade e proteção de dados. Por ser um documento oficial do Governo Federal, com foco em aplicativos e linguagem técnica acessível, representou uma das fontes com grande aderência ao escopo deste trabalho.

As Boas Práticas em Segurança da Informação (TCU, 2012), apesar de desatualizadas em alguns aspectos, contribuíram com embasamento institucional para as diretrizes de controle de acesso e gestão de vulnerabilidades do guia. Por serem uma referência oficial direcionada à Administração Pública Federal, a obra reforça que a segurança da informação não é responsabilidade exclusiva das áreas de governança, mas deve estar presente também nas equipes que desenvolvem os sistemas utilizados pelos órgãos públicos.

O Guia do *Framework* de Privacidade e Segurança da Informação (PPSI, 2024) contribuiu principalmente com diretrizes relacionadas à gestão de riscos, classificação da informação, controle de acesso e proteção de dados pessoais, além de reforçar o alinhamento das recomendações às obrigações normativas da LGPD e às instruções normativas do GSI/PR, que são diretamente aplicáveis ao contexto de desenvolvimento de aplicações governamentais.

O OWASP *Secure by Design Framework* (OWASP, 2025b) foi usado como referência técnica para diretrizes de arquitetura e *design* seguro, além de contribuir com recomendações práticas aplicáveis nas fases de codificação e projeto. Sua orientação ao desenvolvedor facilitou a tradução das recomendações em diretrizes concretas e implementáveis.

A ISO/IEC 27001 (ABNT, 2022) forneceu amplo embasamento normativo e técnico, especialmente na categoria de gestão de vulnerabilidades. Por ser um padrão internacionalmente reconhecido, suas recomendações contribuíram para assegurar o alinhamento do guia com boas práticas consolidadas globalmente.

O NIST *Cybersecurity Framework* (NIST, 2024) complementou as demais fontes com uma visão orientada à gestão de riscos e à segurança ao longo do ciclo de vida, contribuindo principalmente para as diretrizes de arquitetura segura e gestão de vulnerabilidades. Sua estrutura baseada em funções (Identificar, Proteger, Detectar, Responder e Recuperar) ajudou a organizar as recomendações de forma coerente com as etapas do SDLC.

Por fim, a tese de Gatto (2024) contribuiu com uma perspectiva acadêmica estruturada sobre a integração de normas e *frameworks* ao desenvolvimento de *software* seguro. Seus achados reforçaram a importância de tratar a segurança desde as etapas iniciais do desenvolvimento e evidenciaram lacunas que o guia proposto neste trabalho buscou endereçar.

6.4 VERIFICAÇÃO DE COBERTURA FRENTE AO OWASP TOP 10

Esta seção tem por objetivo verificar se o conjunto de diretrizes consolidadas nas etapas anteriores oferece cobertura adequada frente às dez categorias de vulnerabilidade mais recorrentes em aplicações reais, conforme listadas no OWASP Top 10 (OWASP, 2025c). Diferentemente das etapas anteriores, que partiram de fontes orientadas a práticas e controles, essa etapa adota uma perspectiva orientada à ameaça: o objetivo não é validar a qualidade das diretrizes individualmente, mas sim identificar se alguma categoria de vulnerabilidade relevante permanece sem endereçamento adequado no guia. Para isso, a verificação foi conduzida, conforme descrita na seção 2.4, em três passos: análise individual de cada categoria do Top 10, confronto com as diretrizes consolidadas nas etapas anteriores e identificação das lacunas remanescentes. Os resultados estão organizados na Quadro 8, que

apresenta, para cada categoria, as diretrizes correspondentes e a situação de cobertura identificada.

Quadro 8 – Matriz de cobertura frente ao OWASP Top 10

(continua)

Categoria	Vulnerabilidade	Diretriz(es) do guia que endereça(m)	Situação
A01: Quebra de controle de acesso	Usuários agem fora de suas permissões; acesso não autorizado a recursos e dados.	- Autenticação forte (MFA); - Controle de acesso baseado em papéis (RBAC); - Princípio do menor privilégio; - Gestão segura de sessões.	Coberta
A02: Configuração de segurança incorreta	Configurações padrão inseguras, permissões excessivas, funcionalidades desnecessárias habilitadas.	- Princípio do menor privilégio; - Redução da superfície de ataque; - Segurança por padrão (security by default).	Parcialmente coberta
A03: Falhas na cadeia de suprimentos de software	Dependências comprometidas, bibliotecas desatualizadas, componentes de terceiros sem verificação de integridade.	- Gestão de dependências e componentes de terceiros; - Redução da superfície de ataque; - Segmentação e isolamento de componentes; - Monitoramento contínuo.	Parcialmente coberta
A04: Falhas criptográficas	Dados em trânsito ou em repouso sem proteção adequada; algoritmos de criptografia obsoletos ou previsíveis.	- Criptografia de dados em repouso e em trânsito; - Comunicação segura via TLS.	Coberta
A05: Injeção	Dados hostis enviados ao interpretador (SQL, OS, LDAP, XSS, CSRF etc.).	- Validação e sanitização de entradas; - Segurança por padrão (security by default).	Coberta

Quadro 8 – Matriz de cobertura frente ao OWASP Top 10

(continua)

Categoria	Vulnerabilidade	Diretriz(es) do guia que endereça(m)	Situação
A06: Design inseguro	Falhas arquiteturais e de design que não podem ser corrigidas apenas por implementação.	- Modelagem de ameaças; - Redução da superfície de ataque; - Segurança por padrão (security by default).	Coberta
A07: Falhas de autenticação	Credenciais fracas, sessões não expiradas, ausência de MFA em contextos sensíveis.	- Autenticação forte (MFA); - Gestão segura de sessões.	Coberta
A08: Falhas na integridade de software ou dados	Pipelines de CI/CD e atualizações sem verificação de integridade; desserialização insegura.	- Identificação e correção de vulnerabilidades.	Diretriz adicionada
A09: Falhas de registro e monitoramento de segurança	Ausência de logs de eventos de segurança; alertas insuficientes para detecção de incidentes.	- Tratamento seguro de erros e logs; - Identificação e correção de vulnerabilidades; - Monitoramento contínuo e alertas.	Coberta
A10: Tratamento inadequado de exceções	Erros não tratados, mensagens de exceção com dados sensíveis, falhas que expõem estado interno.	- Tratamento seguro de erros e logs; - Segurança por padrão (security by default); - Monitoramento contínuo e alertas.	Coberta

Fonte: elaborado pela autora

Das dez categorias analisadas, sete são endereçadas diretamente por diretrizes já presentes no guia, com atuação sobre as causas raiz descritas pelo OWASP Top 10 (OWASP, 2025c). A Quebra de Controle de Acesso (A01) e as Falhas de Autenticação (A07) são cobertas pelas diretrizes de autenticação forte, controle de acesso baseado em papéis, princípio do menor privilégio e gestão segura de sessões que, em conjunto, reduzem tanto a superfície de ataque nos pontos de autenticação

quanto a possibilidade de acesso não autorizado a recursos. As Falhas Criptográficas (A04) são tratadas pelas diretrizes de criptografia de dados em repouso e em trânsito e de comunicação segura via TLS, que prescrevem o uso de algoritmos atuais e canais protegidos. A categoria de Injeção (A05) é coberta pelas diretrizes de validação e sanitização de entradas e de segurança por padrão, que atuam sobre a causa raiz do problema ao exigir validação. O Design Inseguro (A06) é endereçado pelas diretrizes de modelagem de ameaças, redução da superfície de ataque, e segurança por padrão, que atuam diretamente na fase em que essas falhas se originam. As Falhas de Registro e Monitoramento de Segurança (A09) e o Tratamento Inadequado de Exceções (A10) são cobertos pelas diretrizes de tratamento seguro de erros e logs, monitoramento contínuo e de segurança por padrão, que prescrevem o registro de eventos de segurança, a geração de alertas e o comportamento seguro em situações de falha, sem exposição de informações sensíveis.

Duas categorias apresentaram cobertura parcial. A Configuração de Segurança Incorreta (A02) é coberta no nível da aplicação pelas diretrizes de segurança por padrão, menor privilégio e redução de superfície de ataque, mas configurações de infraestrutura e de ambiente de execução estão fora do escopo do guia, o que é registrado como limitação. As Falhas na Cadeia de Suprimentos de Software (A03) são parcialmente endereçadas pelas diretrizes de gestão de dependências, redução da superfície de ataque e segmentação e isolamento de componentes, reduzindo o risco associado ao uso de bibliotecas desatualizadas ou com vulnerabilidades conhecidas. No entanto, a categoria abrange também a verificação de integridade dos artefatos que compõem o *pipeline* de entrega, como a assinatura digital de pacotes, prática que não havia sido identificada nas etapas anteriores e que, nesta etapa, foi avaliada pelos critérios de priorização definidos na seção 2.3. Foi concluído que a sua aplicabilidade prática ao SDLC é limitada no contexto do guia: a verificação de integridade de artefatos depende da configuração do ambiente de *pipeline*, que pode estar fora do controle direto da equipe de desenvolvimento e envolve decisões de infraestrutura que extrapolam o escopo definido na seção 2.3.2.

No conjunto, os resultados desta etapa indicam que o guia oferece cobertura abrangente frente às categorias do OWASP *Top 10 2025*. Das dez categorias analisadas, apenas uma teve sua cobertura complementada pela inclusão de nova diretriz nesta etapa, e duas apresentaram cobertura parcial com lacunas devidamente

justificadas. Esse resultado é expressivo considerando que o guia foi construído a partir de fontes com perspectivas distintas, sem que nenhuma delas tivesse o OWASP Top 10 como referência direta de estruturação. O fato de que as diretrizes consolidadas a partir dessas fontes, quando confrontadas com uma lista orientada à ameaça, cobrem a quase totalidade das categorias mais recorrentes em aplicações reais, sugere que o processo de análise comparativa e priorização adotado neste trabalho foi eficaz em capturar os aspectos de segurança de maior relevância prática. Por fim, cabe destacar que a única adição necessária nesta etapa (a diretriz de “Verificação de integridade de artefatos de *software*”), incluída a partir da identificação da lacuna na A08, reforça a relevância do processo metodológico adotado: o mecanismo de verificação pelo Top 10 cumpriu sua função de identificar o que as etapas anteriores não haviam capturado, e os critérios de priorização garantiram que a inclusão fosse fundamentada com o restante do guia.

6.5 FORMATO E DISPONIBILIZAÇÃO DO GUIA

O guia foi implementado como uma página *web* estática, desenvolvida em HTML, CSS e JavaScript, e disponibilizada publicamente por meio do GitHub Pages. Essa escolha se justifica por três razões principais:

1. **Acessibilidade do formato:** Um guia destinado a desenvolvedores tem mais chances de ser consultado quando está disponível como uma página *web*, acessível diretamente pelo navegador sem necessidade de *download*, instalação ou *login*. Formatos como PDF ou documentos de texto tendem a ser menos práticos no dia a dia de uma equipe de desenvolvimento, em que a consulta rápida durante o trabalho é mais comum do que a leitura de um documento.
2. **Navegabilidade:** A organização do guia por fases do SDLC, com navegação entre seções, filtros por categoria e exemplos de código expansíveis, seria difícil de reproduzir em um documento estático nos formatos citados no item acima. A interface *web* permite que o desenvolvedor acesse diretamente a fase relevante para o momento em que se encontra no ciclo de desenvolvimento, sem precisar percorrer o documento inteiro.
3. **Disponibilização como código aberto (*open source*):** O repositório do guia foi publicado no GitHub sob licença aberta com o objetivo principal de

torná-lo um artefato colaborativo. Qualquer pessoa pode acessar o conteúdo, sugerir alterações, propor novas diretrizes ou adaptá-lo a contextos específicos por meio dos mecanismos nativos da plataforma, como *issues* e *pull requests*. Esse formato é especialmente relevante considerando que o campo da segurança no desenvolvimento de *software* está em constante evolução: ameaças surgem, normas são atualizadas e boas práticas se consolidam ao longo do tempo, tornando a manutenção colaborativa um fator importante para a longevidade do guia. Além disso, disponibilizar o repositório publicamente facilita que trabalhos futuros possam dar continuidade a esta pesquisa.

O código-fonte do guia está disponível em <https://codigos.ufsc.br/j.knabem/diretrizes-seguranca-para-desenvolvedores> e a versão publicada pode ser acessada em <https://juliaknabem.github.io/diretrizes-seguranca-para-desenvolvedores/>.

6.6 INTERFACE E FLUXO DE NAVEGAÇÃO DO GUIA

O guia foi implementado como uma página *web* estática, hospedada via GitHub Pages, e sua interface foi desenvolvida com o objetivo de facilitar a consulta rápida durante o trabalho de desenvolvimento. As decisões de *design* da interface estão diretamente relacionadas à proposta do guia: ser um artefato prático, de acesso imediato, sem exigir instalação ou autenticação. Esta seção descreve como a interface foi organizada e como o fluxo de navegação foi pensado para apoiar esse objetivo.

A tela inicial, apresentada na Figura 2, funciona como o ponto de entrada do guia. Ela exibe o título do artefato e uma breve descrição do seu escopo, seguida por um conjunto de cartões clicáveis, um para cada fase do SDLC abordada no guia (Requisitos, Projeto, Codificação, Testes e Implantação e Manutenção), além de duas seções adicionais: cobertura frente ao OWASP Top 10 2025; e a listagem das fontes utilizadas. Cada cartão apresenta o nome da fase, uma descrição resumida de seus objetivos de segurança e o número de diretrizes disponíveis, permitindo que o desenvolvedor identifique rapidamente em qual fase se encontra e acesse diretamente o conteúdo relevante.

Figura 2 – Tela inicial do guia

— TCC · JULIA KNABEM · UFSC

Guia de segurança no desenvolvimento de aplicações governamentais

Diretrizes práticas organizadas por etapa do ciclo de vida de desenvolvimento de software (SDLC) para auxiliar desenvolvedores na construção de aplicações governamentais mais seguros, com embasamento em normas, frameworks e legislação vigente.

início

Selecione uma etapa do ciclo de vida de desenvolvimento de software para visualizar as diretrizes de segurança aplicáveis. As diretrizes foram selecionadas a partir de análise comparativa de normas, frameworks e legislação (ISO 27001, NIST CSF, OWASP, LGPD e guias do Governo Federal), priorizadas por impacto, aplicabilidade ao SDLC, relevância normativa e adequação ao contexto de aplicações governamentais.

ETAPAS DO SDLC

01 Requisitos Privacidade, minimização de dados e consentimento 4 diretrizes	02 Projeto Modelagem de ameaças, arquitetura segura e princípios de design 4 diretrizes	03 Codificação Autenticação, criptografia, validação e codificação segura 9 diretrizes
04 Testes Testes de segurança estáticos, dinâmicos e de composição 3 diretrizes	05 Implantação e manutenção Dependências, monitoramento, incidentes e integridade 6 diretrizes	

OWASP TOP 10

Cobertura frente ao OWASP Top 10 verificação →
Mapeamento das diretrizes do guia às 10 categorias de vulnerabilidade mais críticas de 2025

REFERÊNCIAS

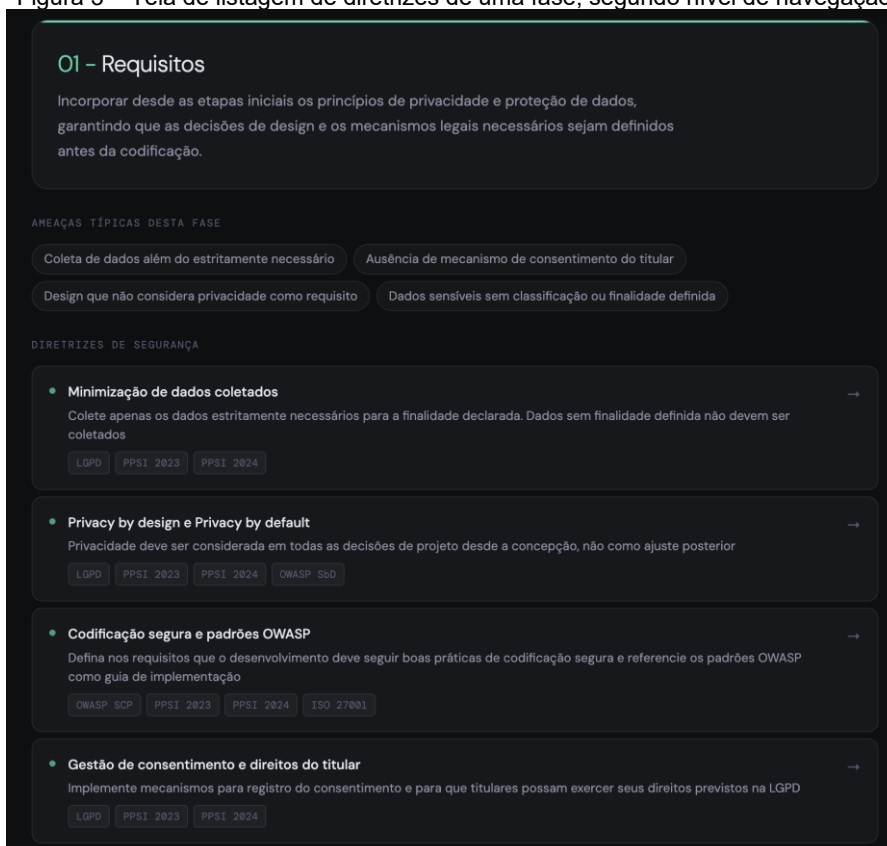
Fontes utilizadas referências →
Normas, legislação, frameworks e guias que fundamentam as diretrizes deste guia

Guia de diretrizes de segurança · TCC · UFSC · 2026 Fontes utilizadas ~

Fonte: elaborado pela autora

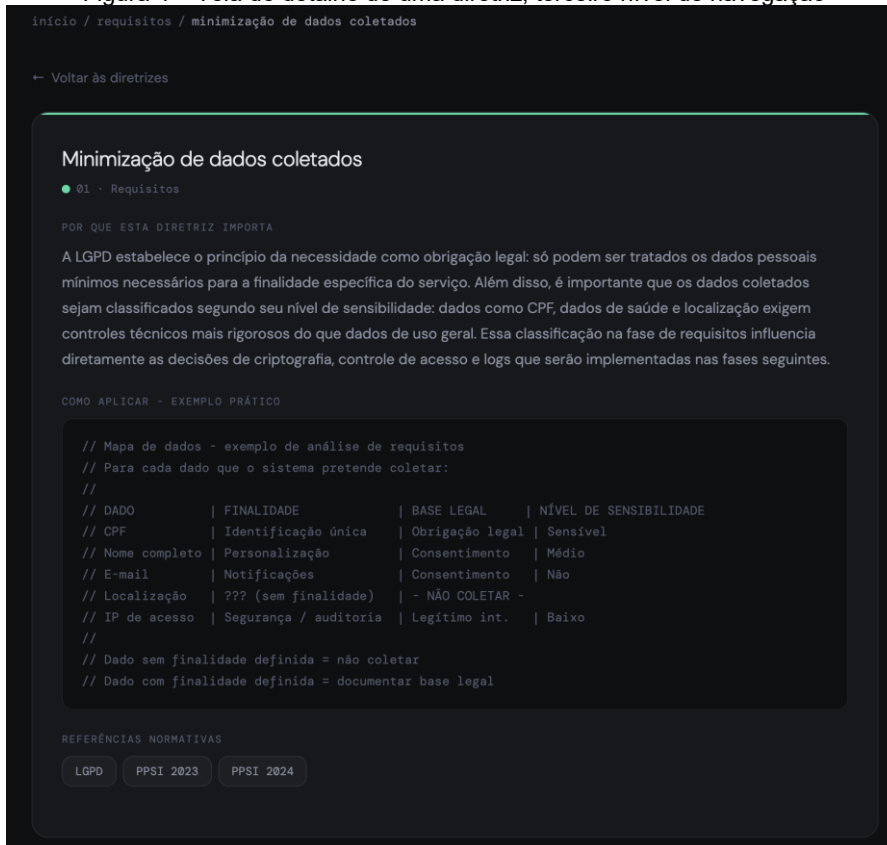
O fluxo de navegação foi estruturado em três níveis hierárquicos. No primeiro nível, o usuário visualiza o mapa geral com as fases do SDLC. Ao clicar em uma fase, é apresentado o segundo nível, que lista as diretrizes associadas àquela etapa (Figura 3). Cada diretriz é exibida como um item clicável que contém o nome da diretriz, uma descrição sucinta e as referências normativas que a fundamentam, representadas por tags com as siglas das fontes. Ao selecionar uma diretriz específica, o usuário acessa o terceiro nível, que detalha a prática de segurança com a descrição completa, a justificativa da sua inclusão no guia, exemplos práticos de implementação com trechos de código e a fase do SDLC à qual pertence (Figura 4).

Figura 3 – Tela de listagem de diretrizes de uma fase, segundo nível de navegação



Fonte: elaborado pela autora

Figura 4 – Tela de detalhe de uma diretriz, terceiro nível de navegação



Fonte: elaborado pela autora

Além das telas das fases do SDLC, a interface disponibiliza duas seções acessíveis diretamente da tela inicial: a seção de cobertura frente ao OWASP *Top 10* 2025 (Figura 5) e a seção de fontes utilizadas. A primeira apresenta uma tabela com as dez categorias de vulnerabilidade e o mapeamento de cada uma às diretrizes do guia, indicando se a cobertura é completa, parcial ou se a diretriz foi adicionada a partir dessa etapa de verificação. A segunda lista todas as referências normativas que fundamentam o guia, organizadas por categoria (legislação nacional, normas internacionais, *frameworks* de segurança e guias governamentais), com *link* direto para os documentos originais (Figura 6).

Figura 5 – Tela de detalhe de uma diretriz, terceiro nível de navegação

Cobertura frente ao OWASP Top 10 2025

Das dez categorias analisadas, sete são cobertas diretamente pelas diretrizes do guia, duas são parcialmente cobertas com limitações documentadas, e uma teve diretriz adicionada a partir desta etapa de verificação.

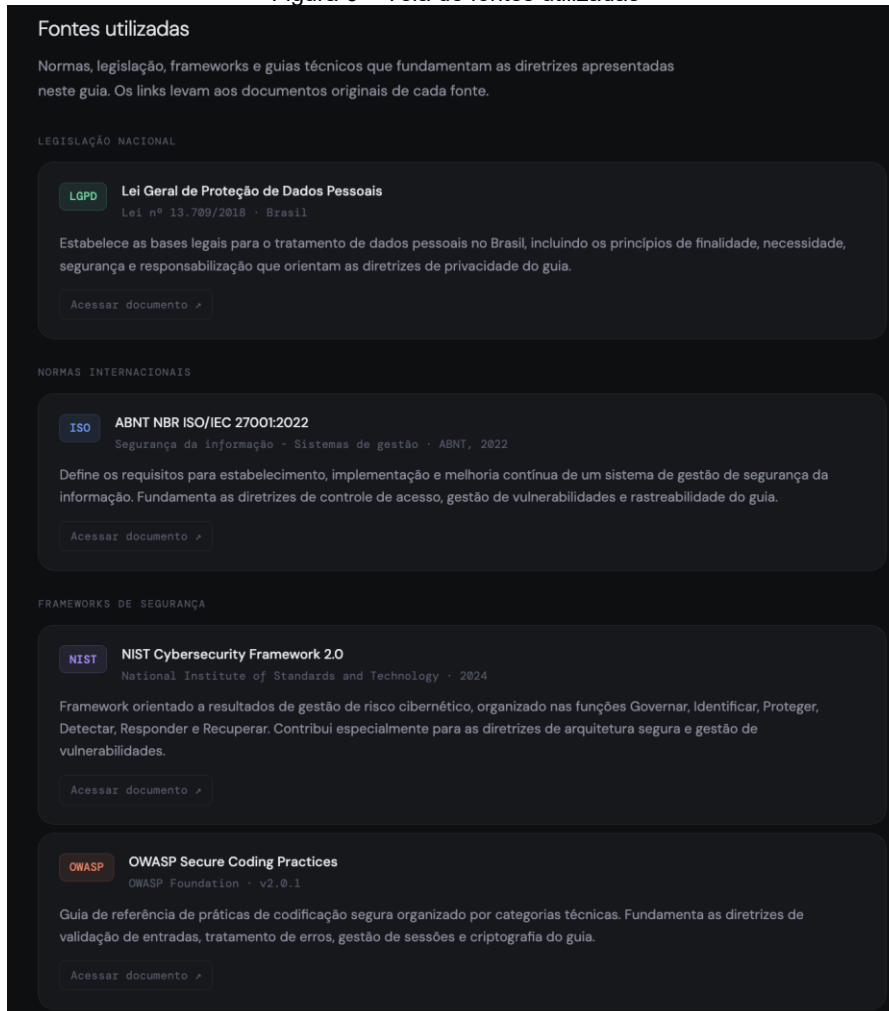
CÓDIGO	VULNERABILIDADE	DIRETRIZES RELACIONADAS	SITUAÇÃO
A01	Quebra de controle de acesso	Autenticação forte (MFA) · RBAC · Menor privilégio · Gestão de sessões	● coberta
A02	Configuração de segurança incorreta	Menor privilégio · Redução da superfície de ataque · Security by default	● parcial
A03	Falhas na cadeia de suprimentos de software	Gestão de dependências (SCA) · Redução de superfície · Segmentação · Monitoramento contínuo	● parcial
A04	Falhas criptográficas	Criptografia em repouso e trânsito · Comunicação segura via TLS	● coberta
A05	Injeção (SQL, XSS, CSRF)	Validação e sanitização de entradas · Security by default	● coberta
A06	Design inseguro	Modelagem de ameaças · Redução de superfície · Security by default	● coberta
A07	Falhas de autenticação	Autenticação forte (MFA) · Gestão segura de sessões	● coberta
A08	Falhas na integridade de software ou dados	Identificação e correção de vulnerabilidades	+ adicionada
A09	Falhas de registro e monitoramento de segurança	Tratamento de erros e logs · Monitoramento contínuo e alertas	● coberta
A10	Tratamento inadequado de exceções	Tratamento de erros e logs · Security by default · Monitoramento contínuo e alertas	● coberta

LIMITAÇÕES DOCUMENTADAS

A02 é parcialmente coberta pois configurações de infraestrutura e ambiente de execução estão fora do escopo do guia. A03 é parcialmente coberta pois a verificação de integridade de artefatos de pipeline depende de configurações de infraestrutura que podem extrapolar o escopo de atuação da equipe de desenvolvimento.

Fonte: elaborado pela autora

Figura 6 – Tela de fontes utilizadas



Fonte: elaborado pela autora

Por fim, a interface foi desenvolvida com responsividade, adaptando-se a diferentes tamanhos de tela. A grade de cartões da tela inicial organiza-se em três colunas em telas largas e se reorganiza em uma única coluna em dispositivos móveis, mantendo a navegabilidade em qualquer contexto de acesso. Esse cuidado com a adaptação de *layout* é importante considerando que desenvolvedores frequentemente consultam dispositivos variados durante o trabalho.

7 AVALIAÇÃO DO GUIA

Esta etapa tem como objetivo verificar se as diretrizes consolidadas ao longo do trabalho oferecem cobertura adequada frente aos requisitos técnicos de segurança do conjunto mínimo definido pelo OWASP *Application Security Verification Standard* (ASVS) 5.0.0, em seu Nível 1.

Conforme descrito na seção 2.5, a avaliação foi conduzida por meio de um mapeamento sistemático entre cada requisito de Nível 1 do ASVS e as diretrizes presentes no guia proposto. O processo seguiu os três passos definidos na metodologia: filtragem dos requisitos pelo Nível 1, confronto de cada requisito com o conjunto de diretrizes, e classificação do resultado como coberto, parcialmente coberto ou não coberto, com indicação da diretriz correspondente nos dois primeiros casos.

O ASVS 5.0.0 é distribuído em 17 capítulos e possui 70 requisitos de Nível 1. Desse total, foram excluídos da análise os 5 requisitos pertencentes ao capítulo V10 (OAuth e OIDC), por tratarem de autorização pelo lado do servidor, o que foge do escopo proposto para o guia desenvolvido. Os capítulos V16 (Segurança de Registro e Tratamento de Erros) e V17 (WebRTC) também não foram incluídos na matriz de correspondência porque nenhum de seus requisitos é classificado como Nível 1 no ASVS 5.0.0 (todos pertencem ao Nível 2 ou 3). Ainda assim, o tema abordado pelo V16 é tratado no guia pelas diretrizes “Tratamento seguro de erros e logs” e “Monitoramento contínuo e alertas”, presentes na fase de Codificação e Implantação, respectivamente. Isso demonstra que o guia proposto incorpora esse controle de forma proativa, ainda que não seja exigido como requisito mínimo pelo padrão de referência adotado na avaliação.

Após essa filtragem, foram analisados os requisitos de Nível 1 dos capítulos pertinentes: V1 (Codificação e Sanitização), V2 (Validação e Lógica de Negócio), V3 (Segurança de *Frontend Web*), V4 (API e *Web Service*), V5 (Tratamento de Arquivos), V6 (Autenticação), V7 (Gerenciamento de Sessão), V8 (Autorização), V9 (*Tokens* autocontidos), V11 (Criptografia), V12 (Comunicação Segura), V13 (Configuração), V14 (Proteção de Dados), e V15 (Segurança de Codificação e Arquitetura).

Os resultados serão discutidos no item a seguir. A matriz de correspondência completa, com o detalhamento de cada requisito e a diretriz correspondente, está disponível no Apêndice B.

7.1 DISCUSSÃO DOS RESULTADOS

Os resultados da avaliação mostram que, dos 65 requisitos de Nível 1 do ASVS analisados, 36 foram classificados como cobertos (55,4%), 16 como parcialmente cobertos (24,6%) e 13 como não cobertos (20%). Considerando os dois primeiros grupos em conjunto, o guia oferece um nível de aderência de, aproximadamente, 80% dos requisitos mínimos do Nível 1 definidos pelo padrão, o que indica uma boa correspondência entre as diretrizes consolidadas ao longo do trabalho. A Tabela 1 apresenta a quantidade de itens por classificação de cada capítulo do ASVS, permitindo uma leitura mais granular dos capítulos mais abordados, dos pontos fortes e das lacunas do guia.

Tabela 1 – Ranking de capítulos do ASVS por cobertura do guia

Posição	Capítulo	Coberto	Parcial	Não coberto	% Cobertura plena
1º	V11 Criptografia	3/3	0	0	100%
1º	V12 Comunicação Segura	3/3	0	0	100%
1º	V13 Configuração	1/1	0	0	100%
1º	V15 Codificação e Arquitetura Seguras	3/3	0	0	100%
2º	V7 Gerenciamento de Sessão	5/6	1	0	83,30%
3º	V8 Autorização	3/4	1	0	75%
3º	V9 Tokens Autocontidos	3/4	1	0	75%
4º	V1 Codificação e Higienezação	4/8	3	1	50%
4º	V2 Validação e Lógica de Negócio	2/4	1	1	50%
4º	V14 Proteção de Dados	1/2	1	0	50%
5º	V6 Autenticação	5/13	4	4	38,50%
6º	V3 Segurança de Web Frontend	3/8	4	1	37,50%
7º	V4 API e Serviços Web	0/2	0	2	0%
8º	V5 Tratamento de Arquivos	0/4	0	4	0%

Fonte: elaborado pela autora

7.1.1 Áreas com cobertura satisfatória

Os capítulos com maior cobertura foram V11 (Criptografia), V12 (Comunicação Segura), V13 (Configuração) e V15 (Codificação e Arquitetura Seguras), todos com 100% dos requisitos analisados classificados como cobertos. Na sequência, V7 (Gerenciamento de Sessão) atingiu 83,3% de cobertura total, e V8 (Autorização) e V9 (Tokens Autocontidos) alcançaram 75% cada, ambos sem nenhum

requisito não coberto. Esses seis capítulos concentram os controles mais diretamente endereçados pelas diretrizes na etapa de codificação do guia, como criptografia de dados em repouso e em trânsito, comunicação via TLS, gestão segura de sessões e controle de acesso baseado em papéis (RBAC). A boa cobertura nessas áreas é coerente com o processo de construção do guia: elas correspondem às categorias que apresentaram alta recorrência nas fontes analisadas na etapa comparativa e que receberam detalhamento técnico mais aprofundado durante a elaboração do guia.

7.1.2 Áreas parcialmente cobertas

Os capítulos V1 (Codificação e Higienização), V2 (Validação e Lógica de Negócio) e V14 (Proteção de Dados) apresentaram cobertura plena em 50% dos seus requisitos, com os demais distribuídos entre parcialmente cobertos e, no caso de V1 e V2, com um requisito não coberto cada. O capítulo V3 (Segurança de Web Frontend) teve o menor percentual entre os capítulos com ao menos alguma cobertura, com apenas 37,5% de requisitos plenamente cobertos e metade deles classificada como parcialmente coberta.

Nesses casos, as diretrizes do guia cobrem os princípios gerais, mas não detalham controles específicos exigidos pelo ASVS. Em V6 (Autenticação), por exemplo, que apresentou 38,5% de cobertura plena (o menor percentual entre os capítulos com cobertura mista), o guia trata de proteção contra força bruta e verificação de senhas comuns, mas não especifica o requisito mínimo de verificação contra as top 3.000 senhas mais comuns definido pelo ASVS, nem detalha o comportamento esperado durante a troca de senha.

Essas lacunas parciais podem ser esperadas em um guia de boas práticas com escopo generalista. A decisão de cobrir os princípios de forma acessível, em vez de detalhar cada controle técnico individualmente, mantém-se coerente com o objetivo do trabalho: oferecer orientações acionáveis para equipes com maturidade variável em segurança, sem transformar o guia em um *checklist* exaustivo. Ainda assim, os capítulos V3 e V6 sinalizam as áreas em que aprofundamentos futuros trariam o maior ganho de cobertura em relação ao padrão.

7.1.3 Áreas não cobertas

Os capítulos V4 (API e Serviços Web) e V5 (Tratamento de Arquivos) foram os únicos com cobertura zero, representando em conjunto 6 dos 13 requisitos não cobertos na validação. No caso de V5, os quatro requisitos não cobertos tratam de validação de tamanho de *upload*, verificação de tipo de arquivo por *magic bytes*, armazenamento seguro de arquivos e proteção contra *path traversal*: temas não abordados em nenhuma diretriz do guia. Essa ausência é parcialmente explicada pelo foco do guia em controles mais transversais ao SDLC, mas o tratamento seguro de arquivos é um vetor relevante em aplicações que permitem envio de documentos (cenário comum em serviços governamentais) e merece ser incluído em versões futuras. Em V4, os dois requisitos não cobertos são a especificação de *Content-Type* em respostas HTTP (V4.1.1) e o uso de WebSocket sobre TLS (V4.4.1), ambos ausentes no guia por tratarem de protocolos e configurações de comunicação que não foram contemplados no escopo original do trabalho.

Os demais 7 requisitos não cobertos estão distribuídos nos capítulos V1 (configuração segura de *parsers* XML para prevenir XXE), V2 (validação de lógica de negócio sequencial), V3 (proteção com mecanismo CORS *preflight*) e V6 (três requisitos de UX de segurança de senhas, como mascaramento do campo com *type* = *'password'*, permissão para *paste* e gerenciadores de senha, e proibição de truncar senhas antes da verificação, além da proibição de perguntas secretas como fator de autenticação). A ausência dessas orientações poderia ser incorporada ao guia sem ampliar significativamente seu escopo.

7.1.4 Diretrizes sem citação ao longo da avaliação

Uma análise complementar dos resultados permite identificar quais diretrizes do guia não foram referenciadas em nenhum dos 65 requisitos analisados. Esse é o caso das diretrizes da fase de Testes, “Minimização de dados coletados”, “Tratamento seguro de erros e logs”, e, com exceção de “Gestão de dependências e componentes de terceiros (SCA)”, nenhuma outra diretriz da etapa Implementação e manutenção foi citada. A ausência dessas diretrizes na matriz de correspondência não indica que sejam desnecessárias, todas elas atenderam aos critérios de priorização relevantes e

endereçam riscos reais ao longo do ciclo de desenvolvimento. Essa divergência é esperada considerando que o ASVS é um padrão voltado à verificação técnica de controles implementados na aplicação, não à avaliação de processos de desenvolvimento ou de capacidade de resposta operacional. As diretrizes do guia não correspondentes aos requisitos do ASVS costumam atuar em uma camada anterior ou posterior à aplicação em si, e por isso escapam ao escopo do instrumento de validação utilizado. Isso reforça a complementaridade entre o guia proposto e o ASVS: enquanto o padrão verifica controles técnicos presentes na aplicação, o guia cobre também o processo de desenvolvimento que precede e sustenta esses controles.

7.1.5 Considerações gerais

De forma geral, os resultados da avaliação indicam que o guia proposto oferece cobertura razoável frente ao conjunto mínimo de requisitos técnicos do ASVS 5.0.0. A distribuição apresentada na Tabela 1 evidencia que as áreas de maior maturidade técnica do guia correspondem justamente aos capítulos com grandes índices de cobertura plena, enquanto as lacunas mais expressivas concentram-se em temas que estavam fora do escopo original do trabalho, como o tratamento de arquivos, ou em controles com alto grau de especificidade técnica, como os requisitos de UX de segurança de senhas. As lacunas identificadas fornecem um mapeamento objetivo das áreas que poderiam ser priorizadas em versões futuras do guia ou em trabalhos que deem continuidade a esta pesquisa.

8 CONCLUSÃO

Este trabalho teve como objetivo propor um guia de diretrizes de segurança voltado para desenvolvedores de aplicações governamentais, organizado segundo as fases do ciclo de vida de desenvolvimento de software (SDLC). Para isso, foi conduzida uma revisão da literatura que reuniu documentos normativos nacionais, *frameworks* internacionais e publicações acadêmicas relacionadas ao desenvolvimento seguro, combinando perspectivas legais, técnicas e operacionais em um único artefato de consulta.

A partir dessa revisão, foram identificadas diretrizes aplicáveis às fases de requisitos, projeto, codificação, testes e implantação e manutenção. A seleção foi guiada por critérios de priorização que levaram em conta o impacto na segurança, a aplicabilidade prática ao SDLC, a relevância normativa e a recorrência entre as fontes, e a adequação ao contexto de aplicações governamentais. O guia resultante foi disponibilizado como um artefato independente, publicado em repositório aberto, com interface *web* que permite consulta direta por fase do desenvolvimento, sem exigir a leitura integral deste trabalho.

A avaliação do guia foi conduzida por meio de um mapeamento sistemático com os requisitos de Nível 1 do OWASP *Application Security Verification Standard* (ASVS) 5.0.0, instrumento selecionado justamente por não ter sido utilizado em nenhuma das etapas de construção do guia, garantindo a independência necessária para que a avaliação não fosse circular. Dos 65 requisitos analisados na matriz de correspondência (após a exclusão dos capítulos fora do escopo definido), 36 foram classificados como cobertos (55,4%), 16 como parcialmente cobertos (24,6%) e 13 como não cobertos (20%). A análise por capítulo, apresentada na Tabela 1, evidencia que os temas com maior cobertura plena foram criptografia (V11), comunicação segura (V12), configuração (V13) e codificação e arquitetura seguras (V15), todos com 100% de cobertura, seguidos de gerenciamento de sessão (V7), com 83,3%, autorização (V8) e *tokens* autocontidos (V9), ambos com 75%. Os capítulos com menor cobertura foram API e Serviços *Web* (V4) e Tratamento de Arquivos (V5), com 0% de requisitos plenamente cobertos, seguidos de autenticação (V6) e segurança de *frontend web* (V3), com 38,5% e 37,5%, respectivamente.

Os resultados indicam que o guia proposto oferece cobertura adequada frente ao padrão mínimo de segurança definido pelo ASVS para aplicações em geral. Ao

mesmo tempo, o processo de verificação evidenciou que as lacunas identificadas não representam falhas na construção do guia, mas sim limitações de escopo conscientes e coerentes com o perfil de uso para o qual ele foi desenvolvido: equipes de desenvolvimento com maturidade variável em segurança, atuando no contexto de aplicações governamentais. De forma complementar, a verificação de cobertura frente ao OWASP Top 10 2025, conduzida durante a etapa de construção, demonstrou que as dez categorias de vulnerabilidade mais prevalentes na prática de desenvolvimento foram endereçadas pelo guia, com cobertura parcial apenas nas categorias cujas lacunas envolvem aspectos de infraestrutura que extrapolam o escopo definido para o trabalho.

Algumas limitações, no entanto, merecem ser destacadas. O mapeamento com o ASVS oferece uma avaliação técnica independente da cobertura do guia, mas não permite verificar se as diretrizes são percebidas como claras ou aplicáveis por quem as utilizaria na prática. Por esse motivo, a validação empírica com desenvolvedores permanece como sugestão de continuação mais direta deste trabalho. Estudos nessa direção poderiam avaliar aspectos como a clareza das diretrizes, a facilidade de integração ao processo de desenvolvimento já adotado pela equipe e a percepção de utilidade em situações reais. Adicionalmente, uma extensão relevante deste trabalho seria a incorporação de legislações internacionais de proteção de dados, como o Regulamento Geral sobre a Proteção de Dados (GDPR) da União Europeia, ampliando o escopo do guia para além do ordenamento jurídico brasileiro. Essa adaptação seria especialmente útil para equipes que desenvolvem aplicações com base de usuários em outros países ou que precisem atender a múltiplos regimes regulatórios simultaneamente. Por fim, a melhoria para cobertura do Nível 1 e ampliação da verificação para os Níveis 2 e 3 do ASVS representam outras frentes relevantes para trabalhos futuros, endereçando controles mais rigorosos para aplicações de maior criticidade.

REFERÊNCIAS BIBLIOGRÁFICAS

Associação Brasileira de Normas Técnicas. ABNT NBR ISO/IEC 27001:2022 – **Segurança da Informação, segurança cibernética e proteção à privacidade – Sistemas de gestão da segurança da informação – Requisitos**. Versão corrigida 31.03.2023. Rio de Janeiro: ABNT, 2022.

Associação Brasileira de Normas Técnicas. ABNT NBR ISO/IEC 27002:2022 – **Segurança da Informação, segurança cibernética e proteção à privacidade – Controles de segurança da informação**. Terceira edição. Rio de Janeiro: ABNT, 2022.

AWS, “O que é uma framework em programação e engenharia?” Disponível em: <https://aws.amazon.com/pt/what-is/framework/>. Acesso em 07 novembro 2025.

BALOGUN, A. O. *et al.* **Software defect prediction using ensemble learning: an ANP based evaluation method**. *FUOYE Journal of Engineering and Technology*, v. 3, i. 2, p.50-55, 2018. DOI: 10.46792/fuoyejt.v3i2.200.

BRASIL. Tribunal de Contas da União. **Auditoria realizada em 2024 apontou irregularidades em descontos do INSS e determinou medidas para corrigir falhas**. Brasília, DF: TCU, 2024. Disponível em: <https://portal.tcu.gov.br/imprensa/noticias/auditoria-realizada-em-2024-apontou-irregularidades-em-descontos-do-inss-e-determinou-medidas-para-corrigir-falhas>. Acesso em: 17 maio 2026.

DEOGUN, Daniel; JOHNSON, Dan Bergh; SAWANO, Daniel. **Secure by Design**. 1. ed. Shelter Island: Manning Publications, 2019.

GATTO, D. D. de O. **Arquitetura integrada de normas e frameworks de desenvolvimento de software seguro aplicada para apoiar a identificação de falhas na especificação de requisitos**. 2024. 293 f. Tese (Programa de Pós-Graduação em Informática e Gestão do Conhecimento) - Universidade Nove de Julho, São Paulo.

Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis. Programa de Privacidade e Segurança da Informação (PPSI). Versão 2.0. Brasília, 2023. Disponível em: https://www.gov.br/governodigital/pt-br/privacidade-e-seguranca/ppsi/guia_requisitos_minimos_apps.pdf. Acesso em: 27 agosto 2025

Guia do Framework de Privacidade e Segurança da Informação. Programa de Privacidade e Segurança da Informação (PPSI). Versão 1.1.4. Brasília, 2024. Disponível em: https://www.gov.br/governodigital/pt-br/privacidade-e-seguranca/ppsi/guia_framework_psi.pdf. Acesso em: 28 agosto 2025.

IBM, “O que é o NIST cybersecurity framework?” Disponível em: <https://www.ibm.com/br-pt/think/topics/nist>. 2025. Acesso em: 29 setembro 2025.

KARIM, N. S. A. *et al.* **The practice of secure software development in SDLC: an**

investigation through existing model and a case study. Security and Communication Networks, v. 9, n. 18, p. 5333-5345, 2016. DOI: 10.1002/sec.1700.

LGPD - Lei Geral de Proteção de Dados Pessoais - Lei nº 13.709, de 14 de agosto de 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014. Diário Oficial da União: seção 1, Brasília, DF, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 03 outubro 2025.

NAZIR, N.; NAZIR, M. K. **A review of security issues in SDLC.** American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), v. 46, n. 1, p. 247-259, 2018.

NIST - National Institute of Standards and Technology. **NIST Cybersecurity Framework 2.0: Resource and Overview Guide.** 2024. Disponível em: <https://doi.org/10.6028/NIST.SP.1299>. Acesso em: 03 outubro 2025.

OKESOLA, O. J. *et al.* **Software requirement in iterative SDLC model.** *Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1 9.* Springer International Publishing, p. 26-34., 2020. DOI: 10.1007/978-3-030-51965-0_2

OWASP. **OWASP Application Security Verification Standard.** Versão 5.0.0. Disponível em: https://github.com/OWASP/ASVS/raw/v5.0.0/5.0/OWASP_Application_Security_Verification_Standard_5.0.0_en.pdf. 2025a. Acesso em: 27 maio 2026.

OWASP. **Melhores Práticas de Codificação Segura OWASP Guia de Referência Rápida OWASP Secure Coding Practices.** Disponível em: <https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/>. 2022. Acesso em: 10 setembro 2025.

OWASP. **OWASP Secure By Design Framework.** Disponível em: <https://owasp.org/www-project-secure-by-design-framework/>. 2025b. Acesso em: 29 setembro 2025.

OWASP. **OWASP Top 10 2025:RC1.** Disponível em: https://owasp.org/Top10/2025/0x00_2025-Introduction/. 2025c. Acesso em: 30 setembro 2025.

RINDELL, K. *et al.* **Security in agile software development: A practitioner survey.** Information and Software Technology, v. 131, p. 106488, 2021. DOI: 10.1016/j.infsof.2020.106488.

RODRIGUES, João Paulo de S.; CORDOVIL, Kleverton Robson da Silva; OHASHI JUNIOR, Orlando Shigueo; TORRES, Renato Hidaka. **Investigação das práticas proativas de desenvolvimento de software seguro adotadas por pequenos times de desenvolvimento.** In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE (SEMISH), 48. , 2021, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira

de Computação, 2021. p. 241-250. ISSN 2595-6205. DOI: <https://doi.org/10.5753/semish.2021.15828>.

SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson, 2019.

STALLINGS, William. **Criptografia e Segurança de Redes**. 6. ed. São Paulo: Pearson, 2015.

Tribunal de Contas da União. **Boas práticas em segurança da informação**. 4. ed. Brasília: TCU, Secretaria de Fiscalização de Tecnologia da Informação. Disponível em: <https://www.tceto.tc.br/profissaogestor/images/governancaTI/2511466.pdf>. 2012. Acesso em: 20 setembro 2025.

ZYAHANA, Felipe Kenji; CUSTÓDIO, Anna Beatriz Santos; KNIHS, Everton; LOPES, Fabio Silva; SILVEIRA, Ismar Frango. **Proposta de framework conceitual para a detecção de vulnerabilidades durante o desenvolvimento de programação**. In: CONGRESSO LATINO-AMERICANO DE SOFTWARE LIVRE E TECNOLOGIAS ABERTAS (LATINOWARE), 21. , 2024, Foz do Iguaçu/PR. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024. p. 229-238. DOI: <https://doi.org/10.5753/latinoware.2024.245739>

APÊNDICE A - ARTIGO BASEADO NESTE TRABALHO

Guia de Diretrizes de Segurança para o Desenvolvimento de Aplicações Governamentais

Julia Knabem

Departamento de Informática e Estatística – Curso de Sistemas de Informação
Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

j.knabem@grad.ufsc.br

Abstract. The growing digitalization of Brazilian public services, together with the enforcement of the General Data Protection Law (LGPD), has increased the need for practical guidance on secure software development for government applications. This paper presents the main results of an undergraduate thesis that proposes a security guideline for developers of government applications, organized according to the phases of the Software Development Life Cycle (SDLC). The guideline was built from a comparative analysis of national regulations, international frameworks and academic literature, and was evaluated against the Level 1 requirements of the OWASP Application Security Verification Standard (ASVS) 5.0.0, reaching an overall adherence of approximately 80%.

Resumo. A crescente digitalização dos serviços públicos brasileiros, aliada à entrada em vigor da Lei Geral de Proteção de Dados Pessoais (LGPD), tem evidenciado a necessidade de orientações práticas para o desenvolvimento seguro de aplicações governamentais. Este artigo apresenta os principais resultados de um trabalho de conclusão de curso que propõe um guia de diretrizes de segurança para desenvolvedores de aplicações governamentais, organizado segundo as fases do ciclo de vida de desenvolvimento de software (SDLC). O guia foi construído a partir de uma análise comparativa de normas nacionais, frameworks internacionais e literatura acadêmica, e avaliado frente aos requisitos de Nível 1 do OWASP Application Security Verification Standard (ASVS) 5.0.0, alcançando aderência geral de aproximadamente 80%.

1. Introdução

O avanço da transformação digital no setor público brasileiro tem levado à criação de um número cada vez maior de sistemas e aplicativos voltados à prestação de serviços ao cidadão. Esse cenário, somado à entrada em vigor da Lei Geral de Proteção de Dados Pessoais (LGPD), impõe às equipes de desenvolvimento a responsabilidade de tratar dados pessoais com o devido cuidado técnico e legal. Entretanto, normas, legislações e *frameworks* de segurança costumam ser extensos e de leitura pouco acessível para quem está no dia a dia da programação, o que dificulta a tradução dessas exigências em práticas concretas de desenvolvimento.

Diante dessa lacuna, este trabalho propõe um guia prático de diretrizes de segurança da informação voltado a desenvolvedores de aplicações governamentais, estruturado segundo as fases do ciclo de vida de desenvolvimento de software (SDLC): requisitos, projeto, codificação,

testes e implantação e manutenção. O objetivo é reunir, em um único artefato de consulta rápida, as contribuições complementares de documentos normativos nacionais, *frameworks* internacionais amplamente adotados e publicações acadêmicas sobre desenvolvimento seguro.

Este artigo resume o trabalho de conclusão de curso que deu origem ao guia, apresentando a metodologia empregada, o processo de construção do artefato e os resultados obtidos em sua validação.

2. Metodologia

A pesquisa possui natureza qualitativa, caráter exploratório e descritivo, e classifica-se como aplicada, uma vez que tem como produto final um artefato prático destinado ao uso por equipes de desenvolvimento. O trabalho foi conduzido em etapas sucessivas de revisão da literatura, análise comparativa e consolidação das diretrizes.

A revisão da literatura reuniu três grupos de fontes: (i) documentos normativos e regulatórios nacionais, como a LGPD, o Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis e as Boas Práticas em Segurança da Informação do Tribunal de Contas da União (TCU); (ii) *frameworks* internacionais amplamente reconhecidos, como a ISO/IEC 27001, o NIST *Cybersecurity Framework* e o OWASP *Secure by Design Framework*; e (iii) publicações acadêmicas relacionadas ao desenvolvimento seguro de *software*.

A partir dessas fontes, foi realizada uma análise comparativa que identificou convergências, divergências e lacunas entre as diretrizes propostas por cada documento. As diretrizes resultantes foram priorizadas com base em cinco critérios: impacto na segurança, aplicabilidade prática e aderência ao SDLC, relevância normativa, recorrência entre as fontes analisadas e adequação ao contexto de aplicações governamentais.

Por fim, o OWASP *Application Security Verification Standard* (ASVS) 5.0.0 foi utilizado exclusivamente na etapa de avaliação do guia, e não como fonte de diretrizes, justamente para preservar a independência avaliativa e evitar que a validação fosse circular em relação ao processo de construção.

3. O Guia Proposto

O artefato resultante do trabalho é um guia de diretrizes e boas práticas (e não um *framework*) organizado segundo as cinco fases do SDLC: requisitos, projeto, codificação, testes e implantação e manutenção. Optou-se por essa estrutura por ser conceitualmente reconhecida e por não impor uma metodologia rígida, permitindo que equipes com processos ágeis ou práticas de DevSecOps apliquem as diretrizes de cada fase no momento adequado do próprio ciclo de trabalho.

Foram incluídas apenas recomendações com aplicabilidade direta ao desenvolvimento de *software*, excluindo-se práticas de caráter puramente gerencial ou institucional que não se traduzem em ações concretas para uma equipe de desenvolvimento. Essa delimitação de escopo está alinhada ao critério de aplicabilidade prática definido na metodologia.

3.1. Disponibilização como Artefato Aberto

O guia foi implementado como uma página *web* estática, hospedada via GitHub Pages, com o objetivo de permitir consulta rápida durante o próprio trabalho de desenvolvimento, sem exigir instalação, autenticação ou a leitura integral do trabalho de conclusão de curso. A navegação foi organizada em níveis hierárquicos: um mapa geral com as fases do SDLC, seguido da listagem de diretrizes de cada fase.

O código-fonte do guia foi publicado sob licença aberta (<https://codigos.ufsc.br/j.knabem/diretrizes-seguranca-para-desenvolvedores>), permitindo que qualquer pessoa proponha alterações, sugira novas diretrizes ou adapte o conteúdo a contextos específicos por meio dos mecanismos nativos da plataforma, como issues e pull requests. Essa escolha busca favorecer a manutenção colaborativa do artefato diante da constante evolução das ameaças e das normas de segurança.

4. Avaliação e Resultados

A avaliação do guia foi conduzida por meio de um mapeamento sistemático entre cada requisito de Nível 1 do OWASP ASVS 5.0.0 e as diretrizes propostas, classificando o resultado como coberto, parcialmente coberto ou não coberto. Do total de 70 requisitos de Nível 1 previstos no padrão, foram excluídos da análise os 5 requisitos referentes a OAuth e OIDC, por tratarem de autorização do lado do servidor, fora do escopo do guia, restando 65 requisitos analisados.

Dos 65 requisitos avaliados, 36 foram classificados como cobertos (55,4%), 16 como parcialmente cobertos (24,6%) e 13 como não cobertos (20%). Somando-se os dois primeiros grupos, o guia apresenta uma aderência de aproximadamente 80% em relação ao conjunto mínimo de requisitos definido pelo padrão. A Tabela 1 apresenta a distribuição dos resultados por capítulo do ASVS.

Tabela 1. Cobertura do guia por capítulo do ASVS 5.0.0

Capítulo	Coberto	Parcial	Não coberto	% Cobertura plena
V11 Criptografia	3/3	0	0	100%
V12 Comunicação Segura	3/3	0	0	100%
V13 Configuração	1/1	0	0	100%
V15 Codificação e Arquitetura Seguras	3/3	0	0	100%
V7 Gerenciamento de Sessão	5/6	1	0	83,3%
V8 Autorização	3/4	1	0	75%
V9 Tokens Autocontidos	3/4	1	0	75%
V1 Codificação e Higiênização	4/8	3	1	50%
V2 Validação e Lógica de Negócio	2/4	1	1	50%
V14 Proteção de Dados	1/2	1	0	50%
V6 Autenticação	5/13	4	4	38,5%
V3 Segurança de Web Frontend	3/8	4	1	37,5%
V4 API e Serviços Web	0/2	0	2	0%
V5 Tratamento de Arquivos	0/4	0	4	0%

Os capítulos com melhor desempenho foram Criptografia (V11), Comunicação Segura (V12), Configuração (V13) e Codificação e Arquitetura Seguras (V15), todos com 100% de cobertura plena, seguidos por Gerenciamento de Sessão (V7), Autorização (V8) e Tokens Autocontidos (V9). Esses temas concentram os controles mais detalhados na fase de

codificação do guia, como criptografia de dados em repouso e em trânsito, uso de TLS e controle de acesso baseado em papéis.

As menores coberturas ocorreram em API e Serviços *Web* (V4) e Tratamento de Arquivos (V5), ambos com 0% de cobertura plena, seguidos por Autenticação (V6) e Segurança de Web Frontend (V3). Essas lacunas refletem, sobretudo, temas fora do escopo original do guia, como validação de *upload* de arquivos e configurações específicas de protocolo.

De forma complementar, verificou-se ainda a cobertura do guia frente às dez categorias do OWASP Top 10 2025, todas endereçadas, ao menos parcialmente, pelas diretrizes propostas. Os resultados, em conjunto, indicam que o guia atende de forma satisfatória às categorias centrais de segurança no desenvolvimento, ao mesmo tempo em que aponta oportunidades específicas, como o tratamento seguro de arquivos, para trabalhos futuros.

5. Conclusão

Este trabalho propôs um guia de diretrizes de segurança da informação voltado a desenvolvedores de aplicações governamentais, estruturado segundo as fases do SDLC e disponibilizado como artefato aberto e independente da leitura do trabalho de conclusão de curso que o originou. A avaliação conduzida frente ao OWASP ASVS 5.0.0 indicou aderência de aproximadamente 80% aos requisitos mínimos de segurança do padrão, com cobertura plena nas áreas de criptografia, comunicação segura, configuração e arquitetura seguras.

As lacunas identificadas na validação, sobretudo relacionadas ao tratamento seguro de arquivos e a controles específicos de autenticação, representam oportunidades concretas para a evolução do guia em trabalhos futuros, bem como a realização de uma validação empírica com desenvolvedores, capaz de avaliar a clareza e a aplicabilidade prática das diretrizes propostas em contextos reais de desenvolvimento.

Referências

- Brasil. (2018) "Lei Geral de Proteção de Dados Pessoais (LGPD)", Lei nº 13.709, de 14 de agosto de 2018.
- ISO/IEC. (2022) "ISO/IEC 27001:2022 – Segurança da informação, segurança cibernética e proteção à privacidade", ABNT.
- NIST. (2024) "The NIST Cybersecurity Framework (CSF) 2.0", National Institute of Standards and Technology.
- OWASP. (2024) "OWASP Secure by Design Framework", Open Web Application Security Project.
- OWASP. (2025a) "OWASP Application Security Verification Standard (ASVS) 5.0.0", Open Web Application Security Project.
- OWASP. (2010) "OWASP Secure Coding Practices – Quick Reference Guide", Open Web Application Security Project.
- Tribunal de Contas da União. "Boas Práticas de Segurança da Informação", TCU.

Gabinete de Segurança Institucional da Presidência da República. "Guia do Framework de Privacidade e Segurança da Informação (PPSI)", GSI/PR.

Knabem, J. (2026) "Guia de Diretrizes de Segurança para o Desenvolvimento de Aplicações Governamentais", Trabalho de Conclusão de Curso, Curso de Sistemas de Informação, Universidade Federal de Santa Catarina (UFSC).

APÊNDICE B - MATRIZ COMPARATIVA DE DIRETRIZES

Diretriz / Aspecto analisado	LGPD (2018)	Guia de Requisitos Mínimos de Privacidade e Segurança da Informação para Aplicativos Móveis (PPSI, 2023)	Boas Práticas em Segurança da Informação (TCU, 2012)	Guia do Framework de Privacidade e Segurança da Informação (PPSI, 2024)	OWASP Secure by Design (2025)	ISO/IEC 27001 (2022)	NIST Cybersecurity Framework (2024)	Arquitetura Integrada de Normas e Frameworks de Desenvolvimento de Software Seguro Aplicada para Apoiar a Identificação de Falhas na Especificação de Requisitos (GATTO, 2024)
Categoria 1 — Gestão de identidade e controle de acesso								
Autenticação e autorização de usuários	•	•	•	•	•	•	•	•
Princípio do menor privilégio	•	•	•	•	•	•	•	•
Gestão de sessões e tokens	•	•	•	•	•	•	•	•
Controle de acesso baseado em papéis (RBAC)	•	•	•	•	•	•	•	•
Categoria 2 — Proteção de dados e privacidade								
Criptografia de dados em repouso e em trânsito	•	•	•	•	•	•	•	•
Minimização e anonimização de dados	•	•	•	•	•	•	•	•
Consentimento e direitos do titular (LGPD)	•	•	•	•	•	•	•	•
Privacy by design e privacy by default	•	•	•	•	•	•	•	•
Armazenamento seguro de credenciais e segredos	•	•	•	•	•	•	•	•
Categoria 3 — Validação de entradas e segurança na implementação								
Validação e sanitização de entradas	•	•	•	•	•	•	•	•
Codificação segura e padrões OWASP	•	•	•	•	•	•	•	•
Tratamento seguro de erros e logs	•	•	•	•	•	•	•	•
Comunicação segura (TLS/HTTPS)	•	•	•	•	•	•	•	•
Categoria 4 — Gestão de vulnerabilidades e atualizações								
Identificação e correção de vulnerabilidades	•	•	•	•	•	•	•	•
Gestão de dependências e componentes de terceiros (SCA)	•	•	•	•	•	•	•	•
Resposta a incidentes de segurança	•	•	•	•	•	•	•	•
Monitoramento contínuo e alertas	•	•	•	•	•	•	•	•
Testes de segurança (SAST, DAST, pentest)	•	•	•	•	•	•	•	•
Continuidade e recuperação de dados (backup, PCN)	•	•	•	•	•	•	•	•
Categoria 5 — Arquitetura e design seguro								
Modelagem de ameaças (Threat Modeling)	•	•	•	•	•	•	•	•
Redução da superfície de ataque	•	•	•	•	•	•	•	•
Segurança por padrão (Secure by Default)	•	•	•	•	•	•	•	•
Segmentação e isolamento de componentes	•	•	•	•	•	•	•	•
Rastreabilidade e auditoria de ações	•	•	•	•	•	•	•	•

APÊNDICE C - MATRIZ DE CORRESPONDÊNCIA COM O ASVS

Requisito ASVS	Descrição Resumida	Classificação	Diretriz(es) do guia que endereça(m)	Justificativa
V1.2 Prevenção de Injeção				
V1.2.1	Verificar que a codificação de saída para resposta HTTP, documento HTML ou XML é adequada ao contexto requerido, para evitar alteração da estrutura do documento.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização cobre explicitamente a codificação de saída em contexto HTML usando sanitize-html e a instrução de usar textContent em vez de innerHTML, alinhada ao requisito de encoding contextual para prevenir XSS.
V1.2.2	Verificar que, ao construir URLs dinamicamente, dados não confiáveis são codificados conforme o contexto e que apenas protocolos URL seguros são permitidos.	Parcialmente coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização aborda a codificação de dados em contextos de saída e cita whitelist nos exemplos, mas não especifica claramente sobre a construção de URLs dinâmicas.
V1.2.3	Verificar que codificação de saída ou escaping é utilizada ao construir conteúdo JavaScript dinamicamente, para evitar JavaScript injection e JSON injection.	Parcialmente coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização cobre explicitamente a codificação de saída ao inserir dados em contexto JavaScript e instrui o uso de sanitize-html, mas não cita explicitamente JavaScript/JSON injection.
V1.2.4	Verificar que queries de banco de dados (SQL, HQL, NoSQL, Cypher) usam queries parametrizadas, ORMs ou mecanismos equivalentes para proteger contra SQL Injection e outras injeções em banco de dados.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização endereça diretamente SQL Injection via queries parametrizadas, com exemplo de uso do módulo pg do Node.js e instrução explícita de nunca concatenar entradas do usuário diretamente em queries.
V1.2.5	Verificar que a aplicação protege contra OS Command Injection e que chamadas ao sistema operacional utilizam queries parametrizadas ou codificação contextual de saída de linha de comando.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização abrange Command Injection como um dos vetores cobertos pelo princípio de nunca passar entradas não confiáveis a interpretadores, incluindo o sistema operacional.
V1.3 Sanitização				
V1.3.1	Verificar que todo input HTML não confiável de editores WYSIWYG ou similares é sanitizado com uma biblioteca de sanitização HTML conhecida e segura ou funcionalidade equivalente do framework.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização instrui explicitamente o uso de sanitize-html para sanitizar entradas de usuários antes de renderizá-las em HTML.
V1.3.2	Verificar que a aplicação evita o uso de eval() ou outros mecanismos de execução dinâmica de código (como Spring Expression Language). Quando não há alternativa, qualquer entrada do usuário incluída deve ser sanitizada antes da execução.	Parcialmente coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização cobre o princípio de nunca executar entradas não confiáveis como código. No entanto, o guia não menciona explicitamente eval() ou mecanismos de execução dinâmica como Spring EL.
V1.5 Desserialização Segura				
V1.5.1	Verificar que parsers XML estão configurados de forma restritiva e que funcionalidades inseguras como resolução de entidades externas estão desabilitadas para prevenir ataques XXE (XML External Entity).	Não coberto	Não consta no guia	O guia não aborda configuração segura de parsers XML nem ataques XXE como diretriz específica.
V2.1 Documentação de Validação e Lógica de Negócio				
V2.1.1	Verificar que a documentação da aplicação define regras de validação de entrada sobre como verificar a validade de cada item de dado em relação a sua estrutura esperada (formatos comuns como CPF, e-mail, telefone ou formatos internos específicos).	Parcialmente coberto	Codificação segura e padrões OWASP (Requisitos)	O guia prevê a definição de requisitos não funcionais de segurança na fase de requisitos, incluindo regras de validação de entrada. No entanto, não exige documentação formal de regras de validação por campo explicitamente.
V2.2 Validação de Entrada				
V2.2.1	Verificar que a entrada é validada para garantir conformidade com expectativas de negócio ou funcionais, usando validação positiva com allowlist de valores, padrões e intervalos, ou comparando com estrutura e limites predefinidos.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz instrui explicitamente: validar tipo, tamanho, formato e intervalo no servidor, usar whitelist de valores esperados.
V2.2.2	Verificar que a aplicação aplica validação de entrada em uma camada de serviço confiável. Embora a validação no cliente melhore a usabilidade e deva ser incentivada, ela não deve ser usada como controle de segurança.	Coberto	Validação e sanitização de entradas (Codificação)	O guia declara explicitamente que "a validação deve ser feita no servidor (nunca apenas no cliente)", com exemplos de middleware de validação server-side.
V2.3 Segurança da Lógica de Negócio				

Requisito ASVS	Descrição Resumida	Classificação	Diretriz(es) do guia que endereça(m)	Justificativa
V2.3.1	Verificar que a aplicação somente processa fluxos de lógica de negócio para o mesmo usuário na ordem sequencial esperada e sem pular etapas.	Não coberto	Não consta no guia	O guia foca em diretrizes técnicas de codificação segura reutilizáveis. A verificação de sequenciamento correto de etapas de fluxo de negócio é específica por domínio de cada aplicação, estando fora do escopo das diretrizes genéricas propostas.
V3.2 Interpretação de Conteúdo Não Intencional				
V3.2.1	Verificar que há controles de segurança para impedir que navegadores renderizem conteúdo ou funcionalidade de respostas HTTP em contexto incorreto. Possíveis controles incluem: validar Sec-Fetch-* headers, usar diretiva sandbox do CSP ou Content-Disposition com tipo attachment.	Parcialmente coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização cobre proteção contra XSS. No entanto, o guia não aborda especificamente os cabeçalhos Sec-Fetch-* ou a diretiva Content-Disposition para controle de contexto de renderização.
V3.2.2	Verificar que conteúdo destinado a exibição como texto simples, e não como HTML renderizado, é tratado com funções de renderização seguras (como createTextNode ou textContent) para evitar execução não intencional de HTML ou JavaScript.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização instrui explicitamente o uso de textContent em vez de innerHTML para inserir dados no DOM do cliente.
V3.3 Setup de Cookies				
V3.3.1	Verificar que cookies possuem o atributo Secure definido e que, se o prefixo __Host- não é usado, o prefixo __Secure- deve ser utilizado no nome do cookie.	Parcialmente coberto	Segurança por padrão (Projeto)	A diretriz de segurança por padrão exemplifica que o cookie deve ser seguro, mas não aborda os prefixos __Host- e __Secure-, que são requisitos específicos do ASVS 5.0 para fortalecimento adicional de cookies.
V3.4 Mecanismos de Cabeçalhos de Segurança do Navegador				
V3.4.1	Verificar que o cabeçalho Strict-Transport-Security está presente em todas as respostas com max-age de pelo menos 1 ano.	Coberto	Comunicação segura via TLS (Codificação)	A diretriz de comunicação segura via TLS inclui exemplo explícito de configuração do header HSTS com max-age=31536000 (1 ano) e includeSubDomains, cobrindo diretamente o requisito.
V3.4.2	Verificar que o cabeçalho CORS Access-Control-Allow-Origin é um valor fixo definido pela aplicação ou, se o valor do header Origin é usado, validado contra uma allowlist de origens confiáveis. Quando Access-Control-Allow-Origin: * for necessário, a resposta não deve conter informações sensíveis.	Parcialmente coberto	DAST - Dynamic Application Security Testing (Testes)	O guia menciona CORS como vetor detectável por DAST, mas não fornece diretriz específica de implementação para configuração de Access-Control-Allow-Origin no allowlist.
V3.5 Separação da Origem do Navegador				
V3.5.1	Verificar que, se a aplicação não usa o mecanismo CORS preflight para prevenir requisições cross-origin indevidas a funcionalidades sensíveis, essas requisições são validadas para garantir que se originam da própria aplicação para defesa contra CSRF.	Coberto	Validação e sanitização de entradas (Codificação)	A diretriz de validação e sanitização cobre explicitamente CSRF via tokens (usando csrf) com exemplo de implementação.
V3.5.2	Verificar que, se a aplicação usa o mecanismo CORS preflight para prevenir uso cross-origin não permitido de funcionalidades sensíveis, não é possível chamar a funcionalidade com uma requisição que não dispare o CORS preflight (verificando Origin e Content-Type ou usando headers extras não CORS-safelisted).	Não coberto	Não consta no guia	O guia não aborda a proteção com o mecanismo CORS preflight.
V3.5.3	Verificar que requisições HTTP a funcionalidades sensíveis usam métodos HTTP adequados (POST, PUT, PATCH ou DELETE), não métodos definidos como "seguros" na especificação HTTP (HEAD, OPTIONS ou GET). Alternativamente, validação estrita dos headers Sec-Fetch-* pode ser usada para garantir que a requisição não originou de chamada cross-origin inadequada.	Parcialmente coberto	Validação e sanitização de entradas (Codificação) + Redução da superfície de ataque (Projeto)	A diretriz de redução da superfície de ataque menciona desabilitar métodos HTTP não utilizados. A validação de métodos HTTP para operações sensíveis é coberta indiretamente, mas o uso de Sec-Fetch-* headers como mecanismo de defesa não é abordado no guia.
V4.1 Segurança Genérica de Serviços Web				
V4.1.1	Verificar que toda resposta HTTP com corpo inclui um header Content-Type que corresponde ao conteúdo real da resposta, incluindo o parâmetro charset para especificar codificação segura (ex.: UTF-8) conforme IANA Media Types.	Não coberto	Não consta no guia	O guia não aborda a validação do corpo da resposta à requisição HTTP.

Requisito ASVS	Descrição Resumida	Classificação	Diretriz(es) do guia que endereça(m)	Justificativa
V4.4 WebSocket				
V4.4.1	Verificar que WebSocket sobre TLS (WSS) é usado para todas as conexões WebSocket.	Não coberto	Não consta no guia	O guia não aborda o uso de WebSocket.
V5.2 Upload de Arquivos e Conteúdo				
V5.2.1	Verificar que a aplicação aceita apenas arquivos de tamanho que possa processar sem causar perda de desempenho ou ataque de negação de serviço.	Não coberto	Não consta no guia	O guia não aborda controles de upload de arquivos como validação de tamanho, limites por usuário ou defesa contra ataques de negação de serviço via uploads excessivos.
V5.2.2	Verificar que, ao aceitar um arquivo (individualmente ou em archive como zip), a aplicação verifica se a extensão corresponde ao tipo esperado e valida que o conteúdo corresponde ao tipo representado pela extensão (incluindo verificação de magic bytes, re-escrita de imagem e bibliotecas especializadas).	Não coberto	Não consta no guia	O guia não possui nenhuma diretriz sobre upload de arquivos. Não há orientação sobre validação de extensão, verificação de magic bytes, re-escrita de imagens ou uso de bibliotecas especializadas para validação de conteúdo de arquivos.
V5.3 Armazenamento de Arquivos				
V5.3.1	Verificar que arquivos enviados ou gerados a partir de entradas não confiáveis e armazenados em pasta pública não são executados como código de programa server-side quando acessados diretamente via requisição HTTP.	Não coberto	Não consta no guia	O guia não possui nenhuma diretriz sobre armazenamento e execução de arquivos.
V5.3.2	Verificar que, ao criar caminhos de arquivo para operações de arquivo, a aplicação usa dados internos ou confiáveis em vez de nomes de arquivo submetidos pelo usuário. Se metadados submetidos pelo usuário precisarem ser usados, validação e sanitização estritas devem ser aplicadas para proteger contra path traversal, LFI, RFI e SSRF.	Não coberto	Não consta no guia	O guia não aborda a construção segura de caminhos de arquivo nem a proteção contra path transversal, LFI, RFI e SSRF.
V6.1 Documentação de Autenticação				
V6.1.1	Verificar que a documentação da aplicação define como controles de rate limiting, anti-automação e resposta adaptativa são usados para defender contra ataques como credential stuffing e força bruta de senha, incluindo como esses controles são configurados para evitar bloqueio malicioso de contas.	Coberto	Autenticação forte - MFA (Codificação) + Modelagem de ameaças (Projeto)	O guia aborda rate limiting e proteção contra força bruta como controles técnicos de autenticação, e na diretriz de modelagem de ameaças fala que as ameaças e decisões precisam ser documentadas.
V6.2 Segurança de Senha				
V6.2.1	Verificar que senhas definidas pelo usuário têm pelo menos 8 caracteres, embora um mínimo de 15 caracteres seja fortemente recomendado.	Coberto	Autenticação forte - MFA (Codificação)	O guia sugere, na exemplificação, a exigência de senhas com mais de 8 caracteres que contenha letras e números.
V6.2.2	Verificar que os usuários podem alterar sua senha.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões aborda explicitamente a troca de senha e a invalidação de sessões após a mudança.
V6.2.3	Verificar que a funcionalidade de troca de senha exige a senha atual e a nova senha do usuário.	Parcialmente coberto	Autenticação forte - MFA (Codificação)	A diretriz de autenticação forte aborda boas práticas de autenticação de forma geral, mas não especifica explicitamente o requisito de solicitar a senha atual durante a troca de senha como controle obrigatório.
V6.2.4	Verificar que senhas submetidas durante o cadastro ou troca de senha são verificadas contra um conjunto de pelo menos as 3.000 senhas mais comuns que correspondam à política de senha da aplicação.	Parcialmente coberto	Autenticação forte - MFA (Codificação)	A diretriz de autenticação forte menciona verificação de senhas comuns, mas não especifica o requisito mínimo de verificação contra as top 3.000 senhas comuns.
V6.2.5	Verificar que senhas de qualquer composição podem ser usadas, sem regras limitando o tipo de caracteres permitidos. Não deve haver exigência de número mínimo de maiúsculas, minúsculas, números ou caracteres especiais.	Parcialmente coberto	Autenticação forte - MFA (Codificação)	O guia sugere senhas com letras e números, mas não define exigências ou limitações de forma explícita.

Requisito ASVS	Descrição Resumida	Classificação	Diretriz(es) do guia que endereça(m)	Justificativa
V6.2.6	Verificar que campos de entrada de senha usam type=password para mascarar a digitação. As aplicações podem permitir que o usuário veja temporariamente a senha completa mascarada, ou o último caractere digitado.	Não coberto	Não consta no guia	O guia não aborda requisitos de UX de segurança para campos de senha, como mascaramento de entrada (type=password) ou a funcionalidade de mostrar/esconder senha.
V6.2.7	Verificar que a funcionalidade de "colar" (paste), auxiliares de senha do navegador e gerenciadores de senha externos são permitidos.	Não coberto	Não consta no guia	O guia não aborda permissão de funcionalidades de UX de senhas como paste e gerenciadores de senha.
V6.2.8	Verificar que a aplicação verifica a senha do usuário exatamente como recebida, sem qualquer modificação como truncamento ou transformação de capitalização.	Não coberto	Não consta no guia	O guia não menciona explicitamente a proibição de truncar ou transformar senhas antes da verificação. Embora o uso de bcrypt/Argon2 preserve integridade da senha, o requisito específico de não modificar a senha recebida não está documentado explicitamente.
V6.3 Segurança Geral de Autenticação				
V6.3.1	Verificar que controles para prevenir ataques como credential stuffing e brute force de senhas estão implementados conforme a documentação de segurança da aplicação.	Parcialmente coberto	Autenticação forte - MFA (Codificação)	A diretriz de autenticação forte inclui proteção contra força bruta com limite de tentativas (ex.: 5 por 10 minutos) e bloqueio temporário, mas não inclui exemplos explícitos de implementação.
V6.3.2	Verificar que contas de usuário padrão (ex.: "root", "admin" ou "sa") não estão presentes na aplicação ou estão desabilitadas.	Coberto	Segurança por padrão (Projeto)	A diretriz de segurança por padrão instrui que novos usuários devem ser criados com papel mínimo e que configurações padrão devem ser as mais seguras possíveis, o que implicitamente cobre a desabilitação de contas padrão privilegiadas.
V6.4 Ciclo de Vida do Fator de Autenticação e Recuperação				
V6.4.1	Verificar que senhas iniciais ou códigos de ativação gerados pelo sistema são aleatoriamente gerados de forma segura, seguem a política de senha existente, expiram em curto período ou após o primeiro uso, e não podem tornar-se a senha permanente.	Coberto	Gestão segura de sessões + Autenticação Forte - MFA (Codificação)	A diretriz de gestão segura de sessões aborda tokens com expiração curta e uso único.
V6.4.2	Verificar que dicas de senha ou autenticação baseada em conhecimento (as chamadas "perguntas secretas") não estão presentes na aplicação.	Não coberto	Não consta no guia	O guia não aborda a proibição de perguntas secretas ou dicas de senha de forma explícita.
V7.2 Segurança Fundamental de Gestão de Sessão				
V7.2.1	Verificar que a aplicação realiza toda verificação de token de sessão usando um serviço backend confiável.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões instrui explicitamente que a verificação de sessões deve ser feita no backend via middleware de autenticação, nunca no cliente.
V7.2.2	Verificar que a aplicação usa tokens auto-contidos ou de referência gerados dinamicamente para gerenciamento de sessão, ou seja, sem usar segredos de API ou chaves estáticas.	Coberto	Gestão segura de sessões + Armazenamento seguro de credenciais e segredos (Codificação)	A diretriz de gestão segura de sessões orienta o uso de JWT com assinatura dinâmica (RS256), e a diretriz de armazenamento seguro proíbe chaves e segredos hardcoded. Tokens estáticos são explicitamente contraindicados.
V7.2.3	Verificar que, se tokens de referência são usados para representar sessões de usuário, são únicos e gerados usando CSPRNG com pelo menos 128 bits de entropia.	Parcialmente coberto	Gestão segura de sessões (03 Codificação)	O princípio de aleatoriedade criptográfica está presente na diretriz de gestão segura de sessões, mas não aborda explicitamente os controles específicos sugeridos neste requisito.
V7.2.4	Verificar que a aplicação gera um novo token de sessão na autenticação do usuário, incluindo reautenticação, e encerra o token de sessão atual.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões aborda explicitamente a geração de novo token após autenticação e invalidação do token anterior, com exemplo de blacklist e função invalidarSessoesDoUsuario.
V7.2 Finalização de Sessão				
V7.4.1	Verificar que, quando a sessão é encerrada (logout ou expiração), a aplicação impede qualquer uso adicional da sessão. Para tokens de referência ou sessões stateful, isso significa invalidar os dados de sessão no backend. Aplicações com tokens auto-contidos precisam de solução como lista de tokens encerrados ou chave de assinatura por usuário.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões cobre explicitamente a invalidação da sessão no logout via blacklist de tokens e a função de invalidar sessões de um usuário específico.
V7.4.2	Verificar que a aplicação encerra todas as sessões ativas quando uma conta de usuário é desabilitada ou excluída.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões aborda a invalidação de todas as sessões ao detectar mudança no estado de segurança do usuário, incluindo desabilitação de conta, com exemplo de invalidarSessoesDoUsuario.
V8.1 Documentação de Autorização				

Requisito ASVS	Descrição Resumida	Classificação	Diretriz(es) do guia que endereça(m)	Justificativa
V8.1.1	Verificar que a documentação de autorização define regras para restringir acesso a nível de função e a dados específicos com base nas permissões do consumidor e nos atributos do recurso.	Parcialmente coberto	Controle de acesso baseado em papéis (RBAC) (Codificação) + Codificação segura e padrões OWASP (Requisitos)	O guia implementa RBAC com exemplos técnicos e define o princípio do menor privilégio, e sugere a documentação na diretriz de codificação segura, mas não exige documentação formal das regras de autorização por função e por dado (data-level).
V8.2 Design de Autorização Geral				
V8.2.1	Verificar que a aplicação garante que o acesso a nível de função é restrito a consumidores com permissões explícitas.	Coberto	Controle de acesso baseado em papéis (RBAC) (Codificação)	A diretriz de controle de acesso baseado em papéis (RBAC) instrui explicitamente que o acesso a cada endpoint deve ser verificado no servidor com base em papel explícito do usuário, com exemplo de middleware requireRole.
V8.2.2	Verificar que a aplicação garante que o acesso a dados específicos é restrito a consumidores com permissões explícitas para aquele item de dado, para mitigar IDOR e BOLA (Broken Object Level Authorization).	Coberto	Controle de acesso baseado em papéis (RBAC) (Codificação) + Princípio do menor privilégio (Projeto)	A diretriz de controle de acesso orienta que usuários só devem acessar os recursos para os quais têm permissão explícita, e o princípio do menor privilégio reforça o acesso mínimo por escopo.
V8.3 Autorização de Nível Operacional				
V8.3.1	Verificar que a aplicação aplica regras de autorização em uma camada de serviço confiável e não depende de controles que um consumidor não confiável possa manipular, como JavaScript do lado do cliente.	Coberto	Controle de acesso baseado em papéis (RBAC) (Codificação)	A diretriz de RBAC instrui explicitamente que papéis e permissões devem ser validados no servidor (do JWT verificado), nunca a partir de dados enviados pelo cliente.
V9.1 Fonte e Integridade do Token				
V9.1.1	Verificar que tokens auto-contidos são validados usando sua assinatura digital ou MAC para proteger contra adulteração antes de aceitar o conteúdo do token.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões instrui explicitamente a validação de JWT via assinatura RS256 antes do uso.
V9.1.2	Verificar que apenas algoritmos de uma allowlist podem ser usados para criar e verificar tokens auto-contidos em determinado contexto. A allowlist não deve incluir o algoritmo "None" e, se tanto algoritmos simétricos quanto assimétricos forem suportados, controles adicionais são necessários para prevenir confusão de chaves.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões orienta o uso exclusivo de RS256 para assinatura de JWT. A rejeição do algoritmo "None" e de algoritmos não permitidos está implicitamente coberta pela instrução de usar apenas RS256 como algoritmo aprovado.
V9.1.3	Verificar que o material de chave usado para validar tokens auto-contidos provém de fontes pré-configuradas confiáveis para o emissor do token, impedindo que atacantes especifiquem fontes e chaves não confiáveis. Para JWTs e outras estruturas JWS, headers como jku, x5u e jwk devem ser validados contra uma allowlist de fontes confiáveis.	Parcialmente coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões aborda validação de JWT com RS256 e fontes de chave confiáveis. No entanto, a validação explícita de headers JWS como jku, x5u e jwk contra uma allowlist não é mencionada no guia, deixando cobertura parcial para esse vetor específico.
V9.2 Conteúdo do Token				
V9.2.1	Verificar que, se um intervalo de validade está presente nos dados do token, o token e seu conteúdo são aceitos apenas se o momento de verificação está dentro desse intervalo. Para JWTs, as claims nbf e exp devem ser verificadas.	Coberto	Gestão segura de sessões (Codificação)	A diretriz de gestão segura de sessões instrui o uso de expiresIn na configuração do JWT e a verificação do timestamp de emissão (iat) para invalidar tokens antigos. A verificação das claims nbf e exp está coberta por esse mecanismo de expiração.
V11.3 Algoritmos de Criptografia				
V11.3.1	Verificar que modos de bloco inseguros (ex.: ECB) e esquemas de padding fracos (ex.: PKCS#1 v1.5) não são utilizados.	Coberto	Criptografia de dados em repouso e em trânsito (Codificação)	A diretriz de criptografia instrui o uso de AES-256-GCM, que é um modo autenticado moderno e seguro. Também sugere a exclusão de métodos inseguros.
V11.3.2	Verificar que apenas cifras e modos aprovados, como AES com GCM, são utilizados.	Coberto	Criptografia de dados em repouso e em trânsito (Codificação)	A diretriz de criptografia especifica explicitamente AES-256-GCM como o algoritmo a ser usado para criptografia de campos sensíveis, com exemplo de código em Node.js.
V11.4 Hashing e Funções Baseadas em Hash				
V11.4.1	Verificar que apenas funções de hash aprovadas são usadas para fins criptográficos gerais, incluindo assinaturas digitais, HMAC, KDF e geração de bits aleatórios. Funções de hash proibidas, como MD5, não devem ser usadas para nenhuma finalidade criptográfica.	Coberto	Criptografia de dados em repouso e em trânsito (Codificação)	A diretriz de criptografia desaconselha explicitamente MD5 e SHA-1 como "equivalentes funcionalmente à ausência de criptografia" e instrui o uso de algoritmos aprovados.
V12.1 Guia de Segurança TLS Geral				

Requisito ASVS	Descrição Resumida	Classificação	Diretriz(es) do guia que endereça(m)	Justificativa
V12.1.1	Verificar que apenas as versões mais recentes recomendadas do protocolo TLS estão habilitadas, como TLS 1.2 e TLS 1.3, com a versão mais recente como opção preferida.	Coberto	Comunicação segura via TLS (Codificação)	A diretriz de TLS especifica explicitamente minVersion: "TLSv1.2", rejeita TLS 1.0 e 1.1, e lista cifras seguras de TLS 1.2 e 1.3, com a versão mais recente como preferida.
V12.2 Comunicação HTTPS com Serviços Externos				
V12.2.1	Verificar que TLS é usado para toda conectividade entre um cliente e serviços HTTP externos, sem fallback para comunicações inseguras ou não criptografadas.	Coberto	Comunicação segura via TLS (Codificação)	A diretriz de TLS orienta que toda comunicação que transmita dados pessoais ou sensíveis deve usar TLS com certificados válidos, incluindo exemplo de redirecionamento HTTP para HTTPS e HSTS.
V12.2.2	Verificar que serviços externos usam certificados TLS publicamente confiáveis.	Coberto	Comunicação segura via TLS (Codificação)	A diretriz de TLS instrui explicitamente que o certificado deve ser emitido por uma CA confiável (ex.: Lets Encrypt) e proíbe certificados autoassinados em produção.
V13.4 Comunicação HTTPS com Serviços Externos				
V13.4.1	Verificar que a aplicação é implantada sem metadados de controle de versão, incluindo as pastas .git ou .svn, ou de forma que essas pastas sejam inacessíveis tanto externamente quanto pela própria aplicação.	Coberto	Armazenamento seguro de credenciais e segredos (Codificação)	A diretriz de armazenamento seguro de credenciais instrui que segredos e metadados sensíveis não devem fazer parte do repositório ou artefatos de build, incluindo o uso de .gitignore e ferramentas de detecção como Gitleaks.
V14.2 Proteção de Dados Geral				
V14.2.1	Verificar que dados sensíveis são enviados ao servidor apenas no corpo ou nos campos de cabeçalho da mensagem HTTP, e que a URL e query string não contêm informações sensíveis como chaves de API ou tokens de sessão.	Coberto	Armazenamento seguro de credenciais e segredos (Codificação) + Segurança por padrão (Projeto)	A diretriz de armazenamento seguro proíbe credenciais e chaves de API no código. A diretriz de segurança por padrão instrui o uso de cookies HttpOnly para tokens. A combinação cobre o requisito de não expor dados sensíveis em query strings.
V14.3 Proteção de Dados do Lado do Cliente				
V14.3.1	Verificar que dados autenticados são removidos do armazenamento do cliente (como o DOM do navegador) após o encerramento do cliente ou da sessão. O header Clear-Site-Data pode ajudar, mas o lado do cliente também deve limpar os dados quando a conexão com o servidor não estiver disponível.	Parcialmente coberto	Segurança por padrão (Projeto)	A diretriz de gestão segura de sessões instrui o armazenamento de tokens em cookies com os atributos HttpOnly e Secure, o que impede o acesso ao token via JavaScript e reduz o risco de exposição no DOM. No entanto, não explicita a orientação de Clear-Site-Data nem a limpeza ativa do lado do cliente.
V15.1 Codificação Segura e Documentação de Arquitetura				
V15.1.1	Verificar que a documentação da aplicação define prazos de remediação baseados em risco para versões de componentes de terceiros com vulnerabilidades e para atualização de bibliotecas em geral, para minimizar o risco desses componentes.	Coberto	Gestão de dependências e componentes de terceiros (SCA) (05 Implantação e manutenção)	A diretriz de gestão de dependências define explicitamente prazos de correção por severidade CVSS (crítica: 24h, alta: 7 dias, média: 30 dias), cobrindo diretamente o requisito de prazo de remediação baseado em risco documentado.
V15.2 Arquitetura Segura e Dependências				
V15.2.1	Verificar que a aplicação contém apenas componentes que não ultrapassaram os prazos de atualização e remediação documentados.	Coberto	Gestão de dependências e componentes de terceiros (SCA) (Implantação e manutenção)	A diretriz de gestão de dependências instrui o monitoramento contínuo de CVEs com npm audit, Snyk e Dependabot, incluindo pipeline CI/CD que bloqueia o build quando há CVE acima do limiar.
V15.3 Codificação Defensiva				
V15.3.1	Verificar que a aplicação retorna apenas o subconjunto de campos necessário de um objeto de dado. Por exemplo, não deve retornar o objeto inteiro, pois alguns campos individuais não devem ser acessíveis aos usuários.	Coberto	Princípio do menor privilégio (Projeto) + Controle de acesso baseado em papéis (RBAC) (Codificação)	O princípio do menor privilégio e o controle de acesso RBAC cobrem o conceito de não expor mais dados do que o necessário.