ChatGPT

Manual de Boas Práticas de Segurança da Informação – ampar.ai

Este guia técnico operacional orienta a equipe de desenvolvimento e DevSecOps da plataforma ampar.ai sobre controles de segurança e compliance. O foco é a proteção de dados sensíveis de saúde e a conformidade com a LGPD (Lei nº 13.709/2018), Resolução CFM 2.314/2022, Lei nº 14.510/2022, RDC ANVISA 657/2022 e normas ISO 27001/27799. A cada etapa do ciclo de desenvolvimento devem ser aplicados princípios de *Privacy by Design* e *by Default*, além de controles técnicos robustos.

1. Governança de Dados Sensíveis de Saúde

- **Classificação e bases legais**: Os dados de saúde são **dados pessoais sensíveis** pela LGPD. Seu tratamento só é permitido com *consentimento específico* do titular ou nas hipóteses legais (ex.: tutela da saúde em atendimento por profissional) [1]. Em telemedicina é obrigatório o consentimento livre e informado do paciente. O processamento deve obedecer aos princípios da LGPD (finalidade, adequação, necessidade, qualidade, transparência, prevenção, segurança etc. Art.6 da LGPD) [2].
- **Medidas técnicas e administrativas**: Conforme LGPD Art. 46, deve-se implementar controles para garantir integridade e sigilo dos dados pessoais de saúde [3]. Isso inclui criptografia de dados em trânsito e em repouso, isolamento de bases de dados, controles de acesso e logs de auditoria. A Resolução CFM 2314/2022 reforça essa exigência em telemonitoramento: "garantia de segurança e confidencialidade, tanto na transmissão quanto no recebimento de dados" [4].
- **Anonimização e eliminação**: Sempre que possível, trate dados em formato anonimizado ou pseudonimizado. Dados anonimizados não são mais pessoais e podem ser usados para análise sem consentimento. Dados pessoais devem ser **eliminados** após o fim do tratamento, salvo obrigações legais. A LGPD (Art.16) determina que dados sejam apagados ao término do tratamento, permitindo-se reter apenas para obrigações legais, estudos (com anonimização) ou uso interno exclusivo (com anonimização) [5]. Defina políticas de retenção claras (por exemplo, histórico médico obrigatório) e procedimentos de descarte seguro.
- **Consentimento e direitos do titular**: Obtenha consentimento destacado sempre que aplicável e registre-o. Atenda prontamente a direitos como acesso, correção e exclusão (Art.18 da LGPD). Mantenha trilhas de auditoria de alterações de dados clínicos. Garanta canais para o titular exercer direitos (consulta de dados, revogação de consentimento, etc.).

2. Segurança em APIs e Microsserviços

- **Autenticação forte e controle de acesso**: Use autenticação por *token* (ex.: JWT) e habilite MFA nos acessos administrativos e médicos. Empregue RBAC (Role-Based Access Control) para limitar privilégios dos serviços e usuários [6]. Toda chamada a API deve verificar escopos/papéis antes de fornecer dados sensíveis.
- **Criptografia de tráfego e dados**: Force TLS 1.3 em todas as conexões de API para proteger dados em trânsito e use cifragem forte (por ex. AES-256) nos dados em repouso [6] [7]. Gere e gerencie chaves criptográficas com HSMs ou gerenciadores seguros (ex.: AWS KMS). Rotacione chaves regularmente e proteja-as com políticas rígidas.
- **Rate limiting e defesa DDoS**: Implemente limites de requisição para evitar ataques de força bruta e negação de serviço [8]. Use API gateways (como Amazon API Gateway, Kong ou NGINX)

para aplicar throttling, filtrar payloads maliciosos e padronizar autenticação. Combine isso com WAF (Web Application Firewall) e proteção em camadas.
- **Validação e segurança de entrada**: Todas as entradas devem ser validadas e sanitarizadas (evitando SQLi, XSS, etc.). Considere usar gateways ou bibliotecas de validação por contrato (OpenAPI+JSON Schema) antes de processar os dados. Mantenha bibliotecas atualizadas e aplique dependabot ou similares para evitar vulnerabilidades conhecidas.

3. Logs Imutáveis e SIEM

- **Logs de auditoria imutáveis**: Use formatos de log que não possam ser alterados (append-only, WORM). Por exemplo, envie logs diretamente para um serviço externo (CloudWatch, ELK, Splunk) configurado com retenção e verificação de integridade [9]. Logs de atividades críticas (login, CRUD em dados sensíveis, alterações de permissões) devem conter timestamp, usuário, serviço, ação e contexto. Não inclua dados pessoais sensíveis nos logs; registre apenas identificadores ou hashes.
- **Centralização e retenção**: Concentre logs de todas as fontes (APIs, aplicações, firewalls, sistemas operacionais) em uma solução SIEM. Defina política de retenção adequada (ex.: $\geq$2 anos) para suportar auditorias e análises forenses [10]. As políticas de retenção devem ser alinhadas a regulamentações (LGPD não especifica prazo exato, mas exige provas de conformidade).
- **Monitoramento e alertas**: Configure alertas para padrões suspeitos (múltiplas falhas de autenticação, transferência de dados em massa, etc.). Ferramentas recomendadas incluem *Elastic Stack* (Elasticsearch/Kibana), Splunk, Graylog ou serviços gerenciados (AWS Security Hub/ Azure Sentinel). Garanta segregação de funções: equipes de segurança e compliance devem ter acesso aos dados de log, distinto dos desenvolvedores, para evitar manipulações maliciosas.

4. Privacy by Design e Privacy by Default

- **Minimização de dados**: Desde o início, colete apenas dados de saúde estritamente necessários. Projetos devem incluir análise de risco de privacidade e *Data Protection Impact Assessments (DPIA)*. Exemplo: não armazene documentos desnecessários (como histórico extra) e desligue logs verbose por padrão se não estritamente requeridos.
- **Configurações padrão protegidas**: Configure componentes (bancos, caches, filas) para o modo mais restritivo por padrão. Por exemplo, APIs novas devem começar sem expor endpoints de debug; recursos críticos (DB, filas de mensagens) devem vir criptografados e sem credenciais embutidas no código.
- **Segurança integrada no desenvolvimento**: Inclua requisitos de privacidade em todas as fases do desenvolvimento (definições de endpoints, contratos de API e interfaces). Cada sprint deve revisar o impacto em privacidade de novas features (ex.: adicionar um campo de dados exige justificativa legal e consentimento adequado).
- **Transparência**: Documente em arquitetura e código como os dados são protegidos. Tenha documentação acessível dos fluxos de dados sensíveis. Se houver consentimentos, mantenha logs da aceitação do usuário. Esses princípios são reforçados pela LGPD (Art.6º VI e VII) que exigem “informações claras” ao titular e uso de “medidas técnicas e administrativas” para proteção [2].

5. Ciclo de Vida Seguro do Software (SDLC)

- **Desenvolvimento seguro (Secure SDLC)**: Adote padrões OWASP e frameworks de codificação segura. Use *SAST* (análise estática de código) em cada commit – por exemplo, SonarQube ou

Snyk – bloqueando builds se forem detectadas vulnerabilidades críticas. Estabeleça revisão de código focada em segurança.

- **Integração Contínua / Entrega Contínua**: Construa pipelines que integrem scanners de segurança. No CI/CD (ex.: GitHub Actions, Jenkins), inclua etapas de SAST, auditoria de dependências (Dependabot, OWASP Dependency-Check) e análise de container (Trivy, Clair) em imagens geradas. Gatilhos de segurança devem interromper o deploy em caso de falhas.
- **Testes de invasão**: Realize testes de penetração periódicos (pentests) e avaliações de vulnerabilidade em ambientes de staging, focando em APIs, UI e infraestrutura cloud. Use também *DAST* e fuzzing (ex.: OWASP ZAP, Burp) em versões release. Incorpore lições nos sprints seguintes.
- **Documentação e compliance**: Documente requisitos de segurança e privacy em histórias de usuário. Mantenha um repositório atualizado de políticas internas (ex.: política de senhas, data flow diagrams). Integre checklists de compliance (consentimento, documentação médica, normas da ANS/CFM) no planejamento do release.

6. Conformidade RDC ANVISA nº 657/2022 (SaMD para IA)

- **Classificação de risco**: A plataforma de IA de teletriagem pode ser considerada *SaMD* (Software as Medical Device). De acordo com a RDC 657, software que previne, diagnostica ou trata condições é regulado [11]. Classifique o risco segundo normas de saúde: consultoria recomenda registar classes III/IV e notificar I/II se não for uso interno. Para SaMD destinados ao público ou operadoras, é prudente cumprir RDC 185/2001 (I/II via notificação, III/IV via registro) e obter AFE (Autorização de Funcionamento de Empresa) da ANVISA [12].
- **Gestão da qualidade e segurança**: Implemente processo de desenvolvimento compatível com normativas médicas (por ex. IEC 62304 para software médico). Mantenha um QMS (Quality Management System) alinhado à ISO 13485. Designe gestor técnico e jurídico, conforme exigido [13]. Para Classes III/IV, terá de obter certificação de Boas Práticas (RDC 665/2022).
- **Registro e documentação técnica**: Em registro técnico (RDC 657) inclua arquitetura detalhada do software, descrição de algoritmos, validação e verificação (V&V), gestão de riscos e evidências clínicas. A ANVISA exige relatório técnico com “arquitetura de cibersegurança e controles” e “estratégias de gerenciamento de risco” usados no SaMD [14]. Inclua informações sobre atualização de software, requisitos mínimos de hardware/software, alertas, interoperabilidade e **segurança cibernética** nos rótulos/documentação do produto [15].
- **Rótulos e instruções de uso**: Mesmo em distribuição digital, inclua no software ou site informações obrigatórias (versão do produto, advertências, procedimentos de atualização, requisitos do sistema e instruções de segurança cibernética) [15]. Mantenha sempre versão de produção registrada com ANVISA se o software se destinar a usuários finais ou profissionais de saúde, e reporte alterações significativas nos algoritmos segundo prazos legais (RDC 743/2022).

7. Hardening de Infraestrutura e Backups

- **Endurecimento de sistemas**: Aplique boas práticas de hardening em servidores, containers e serviços cloud. Use imagens base seguras (CIS Benchmarks) e verifique configurações de rede (firewalls, security groups). Restrinja acessos a portas essenciais (p.ex.: apenas 443/8443 público). Proteja consoles de gestão (AWS/GCP/Azure) com MFA e IP whitelisting. Realize testes de vulnerabilidade nos sistemas operacionais e faça correções rápidas de patches críticos.
- **Controle de acesso (RBAC)**: Defina roles e políticas IAM estritas em cloud (AWS IAM, Azure RBAC). Cada serviço deve ter o mínimo de permissões necessárias. Separe contas: uma para deploy/infraestrutura, outra para produção. Audite regularmente contas e chaves de API não utilizadas. Use ferramentas de gestão de identidade (AWS IAM Access Analyzer, Google Cloud IAM) para detectar excessos de privilégio.

- **Criptografia e chaves**: Habilite criptografia por AES-256 em todos os volumes de disco (EBS, discos gerenciados) e buckets de backup. Use TLS 1.3 para end-to-end do tráfego (incluindo conexões de banco de dados). Gerencie chaves mestras em HSM ou serviços gerenciados (AWS KMS, HashiCorp Vault) com rotação automática. Para criptografia de backup, utilize chaves diferentes das de produção para maior segregação.
- **Backup resiliente**: Estabeleça política de backup diário (p.ex., AWS Backup ou scripts agendados) incluindo bases de dados e volumes críticos. Armazene cópias criptografadas em regiões ou contas separadas (geo-redundância). Realize testes de restauração periodicamente para garantir integridade. Implemente versionamento em bancos (timestamp em objetos de dados) e use estratégias de GFS (Grandfather-Father-Son) para garantir retenção de longo prazo. Mantenha backup offline (armazenamento isolado) para proteger contra ransomware.

8. Procedimentos de Resposta a Incidentes e Notificação à ANPD

- **Plano de resposta a incidentes**: Tenha um **playbook** de resposta a incidentes cibernéticos que inclua: identificação, contenção, erradicação, recuperação e lições aprendidas. Treine o time para ativar o plano imediatamente após detecção (observar Resolução CD/ANPD nº15/2024) [16] . Documente ações de mitigação (como fechamento de brechas, limpeza de sistemas infectados, troca de credenciais) e comunique internamente responsabilidades de cada equipe.
- **Detecção e classificação**: Monitore continuamente com SIEM e sistemas de IDS/IPS (por ex. Falco, GuardDuty) para identificar invasões, vazamentos ou malwares. Classifique o incidente com base no impacto (impacto em dados sensíveis, prejuízos operacionais ou legais). Atue de forma a bloquear imediatamente comunicações maliciosas (revogar chaves, isolar redes).
- **Comunicação à ANPD e titulares**: Em caso de incidente que “acarrete risco ou dano relevante” a dados pessoais, a LGPD exige notificar **imediatamente** a Autoridade Nacional (ANPD) e, quando aplicável, os titulares afetados (Art.48 LGPD). A Resolução CD/ANPD Nº15/2024 exige notificação em até **72 horas** após detecção [17] . Use o formulário oficial CIS da ANPD. Inclua detalhes do incidente (natureza, dados expostos, medidas adotadas) e informe titulares com clareza sobre riscos e providências (troca de senhas, monitoramento de crédito etc).
- **Responsabilidade documental**: Registre o incidente em um registro interno de segurança, com evidências (logs de segurança, dumps), e mantenha disponível para auditoria. A ANPD exige registro detalhado dos eventos e das ações de mitigação [16] . Isso reforça a postura de transparência e facilita análises futuras. Revise lições aprendidas e atualize o plano de resposta conforme necessário.

9. Ferramentas e Arquitetura Segura

- **Integração DevSecOps**: Adote uma pipeline CI/CD que incorpore segurança. Exemplos: usar **GitHub Actions** (ou Jenkins) com runners isolados, configurados para executar SAST (SonarQube), DAST (ZAP), testes de container (Trivy) e verificação de compliance. Utilize GitOps (ArgoCD ou Flux) para provisionar infraestrutura via código (Terraform/CloudFormation). Insira *gates* que bloqueiem a liberação em caso de violação de políticas. Conforme referência, ferramentas chave incluem Vault (gestão de segredos), Falco (monitoramento runtime de contêiner), Trivy/Snyk (varredura de imagens), OPA/Kyverno (políticas declarativas) [18] .
- **Gerenciamento de segredos**: Centralize segredos (senhas, chaves API, certificados) em cofre seguro (HashiCorp Vault, AWS Secrets Manager). Nunca armazene segredos em plain text no código ou em repositórios. Automatize a renovação de certificados TLS (ex.: cert-manager em Kubernetes). Controle acesso ao vault via RBAC e registre auditorias de acesso.
- **Segurança em runtime**: Na arquitetura de microsserviços (Docker/Kubernetes), habilite políticas de segurança de pods (PodSecurityPolicy ou equivalente) para restringir capacidades (por ex. sem escalonamento de privilégios). Use scanners de vulnerabilidade em imagens base e

ferramentas de monitoramento (Falco, Sysdig) para detectar comportamento anômalo em runtime (execução de binários suspeitos, tráfego não autorizado).

- **Monitoramento de integridade**: Implemente verificações de integridade de sistema (tripwire, AIDE) em servidores críticos. Use EDR (Endpoint Detection and Response) em VMs/serviços onde aplicável. Configure *health checks* de segurança na infraestrutura para auto-remediação (containers reiniciam se detectar inconsistências).
- **Arquitetura de rede**: Segmente a rede: isole bancos de dados e serviços críticos em sub-redes privadas sem acesso público. Use gateway de API para tráfego externo e proxys reversos para serviços internos. Acesso SSH/Admin deve passar por jump hosts ou VPN corporativa. Considere Zero Trust: cada componente deve autenticar toda conexão recebida.
- **Ferramentas de observabilidade**: Além do SIEM, monitore métricas e logs de segurança: instale Prometheus/Grafana ou Elastic (ELK) para coletar logs de aplicações e infraestrutura. Analise o comportamento do tráfego (ferramentas IDS). Esses sistemas ajudam a detectar incidentes emergentes e servem como base para auditoria post-mortem.

10. Templates de Checklist de Conformidade por Sprint

Verificação	Critério
SAST/Análise Estática	Executar SonarQube/SAST no novo código. Nenhuma vulnerabilidade crítica/vulnerável aberta deve persistir.
DAST/Testes de Invasão	Concluir varredura DAST (ex.: OWASP ZAP) nas APIs. Nenhum endpoint exposto a ataques conhecidos (SQLi, XSS etc.)
Dependências de Segurança	Rodar scanner de dependências (Dependabot/Snyk). Atualizar bibliotecas com CVEs críticos antes do merge.
Infraestrutura como Código	Revisar templates Terraform/CloudFormation com ferramenta (Checkov/Tfsec). Garantir configurações seguras (ex.: criptografia habilitada, RBAC).
Logs e Monitoramento	Verificar que novos endpoints geram logs de auditoria. Atualizar regras do SIEM. Validar que bancos de dados auditem acessos a dados sensíveis.
Autenticação/ Autorização	Confirmar que novas funções/endpoints estão protegidos (JWT/MFA aplicados). Revisar políticas RBAC após mudanças de acesso.
Hardening e Patches	Aplicar configurações de segurança (CIS) nos novos servidores/ containers. Atualizar imagens base com patches.
Backup & Dados Críticos	Garantir backup diário das novas bases de dados/volumes, criptografado (AES-256). Testar restauração de esquema crítico após mudanças.
Privacidade & LGPD	Validar que coleta de novos dados pessoais tem base legal (consentimento, contrato, etc.). Atualizar contratos/termos de uso conforme necessário.
Documentação & Compliance	Atualizar documentação (data flow, arquitetura de segurança). Incluir evidências de auditoria e aprovação (ex.: revisão de código) no sprint review.

Este checklist deve ser preenchido em cada sprint para garantir que cada entrega cumulativa mantém a conformidade legal e os padrões de segurança exigidos. Ajuste-o conforme a complexidade de cada funcionalidade.

Fontes: legislação e normas vigentes [3] [1] [5] [4] [6] [8] [9] [10] [7] [17] [16] [15] [14] [18] [2]. As recomendações seguem também guidelines reconhecidas (ISO 27001/27799, OWASP, ANVISA).

---

[1] Artigo 11: Tratamento de dados pessoais sensíveis - Capítulo 2 - DO TRATAMENTO DE DADOS PESSOAIS - LGPD Brasil
https://lgpd-brasil.info/capitulo_02/artigo_11

[2] Artigo 6: Princípos que regem as atividades de tratamento - Capítulo 1 - DISPOSIÇÕES PRELIMINARES - LGPD Brasil
https://lgpd-brasil.info/capitulo_01/artigo_06

[3] LGPD | Ministério da Saúde
https://sisaps.saude.gov.br/sistemas/esusaps/docs/manual/LGPD/

[4] RESOLUÇÃO
https://sistemas.cfm.org.br/normas/arquivos/resolucoes/BR/2022/2314_2022.pdf

[5] Artigo 16: Eliminação de dados pessoais - Capítulo 2 - DO TRATAMENTO DE DADOS PESSOAIS - LGPD Brasil
https://lgpd-brasil.info/capitulo_02/artigo_16

[6] [8] 8 Essential API Security Best Practices | Zuplo Blog
https://zuplo.com/blog/2025/01/31/api-security-best-practices

[7] Top 25 Cloud Security Best Practices
https://www.sentinelone.com/cybersecurity-101/cloud-security/cloud-security-best-practices/

[9] Audit Logging: What It Is & How It Works | Datadog
https://www.datadoghq.com/knowledge-center/audit-logging/

[10] Guia de SIEM (Gerenciamento de Informações de Segurança e Eventos) | Veeam
https://www.veeam.com/blog/pt/siem-guide-security-information-event-management.html

[11] Título da Apresentação
https://www.interamericancoalition-medtech.org/regulatory-convergence/wp-content/uploads/sites/4/2022/09/English-Iran-3-SaMD-29-08-2022-Webinar-FDA-Practical-Implementation-of-Conformity-Assessment-RDC-657-2022.pdf

[12] [13] [14] [15] ANVISA Regulation of SaMD in Brazil : RDC 657/2022
https://www.medtechinnovate.io/post/anvisa-regulation-of-samd-in-brazil-rdc-657-2022

[16] [17] Novas regras para comunicação de incidentes de segurança são estabelecidas pela ANPD | TI INSIDE Online
https://tiinside.com.br/29/04/2024/novas-regras-para-comunicacao-de-incidentes-de-seguranca-sao-estabelecidas-pela-anpd/

[18] Adv DevSecOps – Part 7: Deploy the Full DevSecOps Stack with Terraform and GitOps | by DiPAK KNVDL | Jun, 2025 | Medium
https://medium.com/@kdeepak99/part-7-deploy-the-full-devsecops-stack-with-terraform-and-gitops-77023e573a54