PI- 4°SEMESTRE

RISCOS E VULNERABILIDADES DO DASHBOARD CANNOLI

Projeto: Dashboard Interativo (Cannoli)
Turma / Grupo: 4º Semestre (Grupo 02)
Integrantes: Bruna Farias Pires; Anie Augusto Bissoli; Erika Santana da Silva; Luiza Domingues Chaveiro Correia

1. Introdução

Este documento traz um levantamento aprofundado de **riscos, vulnerabilidades e ameaças** relacionadas ao desenvolvimento do *Dashboard Interativo da Cannoli* . O objetivo é entregar, um diagnóstico de segurança suficientemente detalhado para orientar medidas imediatas de mitigação e o planejamento das atividades futuras de segurança (pen tests, hardening, monitoramento).

2. Contexto do sistema

- Usuários: **Administradores Cannoli** e **Clientes/Restaurantes**.
- Funcionalidades sensíveis: autenticação, visualização e exportação de KPIs (CSV/PDF/XLS), integração com APIs externas (pagamento, delivery), armazenamento em nuvem, simulação em tempo real, geração de relatórios.
- Dados: dados pessoais de clientes/restaurantes, métricas operacionais e transacionais, logs de uso e possivelmente dados financeiros. Esses aspectos elevam requisitos legais (LGPD) e de confidencialidade.

3. Levantamento detalhado de riscos

1) Acesso não autorizado / autenticação fraca

- **Descrição:** Usuários não autorizados obtêm acesso administrativo ou de cliente por credenciais roubadas, credenciais fracas ou ausência de MFA.
- **Atores:** atacantes externos, insiders mal-intencionados.
- **Ativos impactados:** base de usuários, dashboards, relatórios, dados pessoais.
- **Consequências:** vazamento de dados, alteração de KPIs, fraude, sanções LGPD.
- **GUT:** G=5, U=5, T=5 → **125** (Prioridade 1)
- **Controles recomendados:**
  - Preventivos: autenticação multifator (MFA), políticas de senha (comprimento, complexidade), bloqueio por tentativas, armazenamento de senhas com hashing forte (bcrypt/argon2), RBAC/least privilege.
  - Detectivos: alertas de login anômalo (IP/geolocalização), alertas de tentativa de brute-force.

    - Corretivos: reset seguro de credenciais, logging completo de sessões.
- **Critério de aceite:** login com MFA obrigatório para contas administrativas; senhas armazenadas por algoritmo moderno; testes de brute-force bloqueados.
- **Responsável / Esforço:** Backend + DevOps; **Esforço:** Médio.

2) Exposição de dados sensíveis / vazamento (em trânsito e em repouso)

- **Descrição:** Dados pessoais (nome, telefone, e-mail) ou dados estratégicos são interceptados ou acessados indevidamente.
- **Atores:** atacantes man-in-the-middle, acesso indevido por configuração incorreta.
- **Ativos impactados:** banco de dados, backups, logs, arquivos exportados.
- **Consequências:** violação de confidencialidade, multas LGPD, perda de confiança.
- **GUT:** G=5, U=4, T=5 → **100** (Prioridade 2)
- **Controles recomendados:**
    - Preventivos: TLS obrigatório (HTTPS/TLS 1.2+), criptografia at-rest (AES-256) para bases e backups, mascaramento de dados sensíveis nas exportações, tokenização quando possível.
    - Detectivos: auditoria de acesso a dados, alertas de exfiltração em grandes volumes.
    - Corretivos: rotação de chaves, revogação de credenciais, comunicação e plano de remediação de incidentes.
- **Critério de aceite:** tráfego sempre em HTTPS; backups criptografados; exportações apenas com consentimento/anonimização quando necessário.
- **Responsável / Esforço:** Dev + DevOps; **Esforço:** Médio.

3) Configuração insegura em nuvem / segredos expostos (buckets públicos, .env no repositório)

- **Descrição:** Serviços em nuvem com permissões amplas, buckets públicos com dados, segredos (API keys) em código.
- **Atores:** atacantes que escaneiam configurações públicas; ex-colaboradores.
- **Ativos impactados:** storage, serviços, integrações com terceiros.
- **Consequências:** vazamento de dados, uso indevido de serviços (faturamento), ataque por cadeia de credenciais.
- **GUT:** G=5, U=4, T=4 → **80**
- **Controles recomendados:**
    - Preventivos: uso de secrets manager; variáveis de ambiente fora do repo; políticas de bucket privadas; princípio do menor privilégio IAM.
    - Detectivos: auditoria de acesso a buckets, alertas de exposição pública.
    - Corretivos: revogação de chaves comprometidas; políticas de rotacionamento.
- **Critério de aceite:** nenhum segredo em repositório; política de IAM aplicada; buckets privados; prova (scan) de ausência de segredos no histórico git.
- **Responsável / Esforço:** DevOps; **Esforço:** Médio.

4) Injeção (SQL/NoSQL/Command injection) e validação insuficiente

- **Descrição:** Entrada do usuário não sanitizada que permite execução de consultas/ comandos maliciosos.
- **Atores:** atacantes externos explorando endpoints.
- **Ativos impactados:** banco de dados, integridade de dados.
- **Consequências:** corrupção/exfiltração de dados, acesso privilegiado.
- **GUT:** G=5, U=4, T=4 → **80**
- **Controles recomendados:**

    - Preventivos: queries parametrizadas / ORM; validação de entrada no servidor; whitelist de campos; uso de prepared statements.
    - Detectivos: WAF (regras para injeção), monitoramento de queries anômalas.
    - Corretivos: restauração de backups, correção de ponto vulnerável.
- **Critério de aceite:** testes automatizados que provem que entradas maliciosas não afetam consultas; SAST/DAST sem falhas críticas relacionadas.
- **Responsável / Esforço:** Backend; **Esforço:** Médio.

5) Insuficiente logging, monitoramento e incapacidade de resposta a incidentes

- **Descrição:** Falta de logs relevantes e de um plano de IR; detecção tardia de incidentes.
- **Atores:** qualquer atacante que consiga persistir tempo no ambiente.
- **Ativos impactados:** todos — aumenta tempo de exposição e danos.
- **Consequências:** demora na contenção, perda de evidências, maior impacto reputacional.
- **GUT:** G=5, U=4, T=4 → **80**
- **Controles recomendados:**
    - Preventivos: arquitetura com logs centralizados (ex: ELK/Splunk/SIEM).
    - Detectivos: alertas sobre padrões anômalos (alto volume de downloads, alterações de roles).
    - Corretivos: playbooks de IR, backups testados e processos de comunicação.
- **Critério de aceite:** logs essenciais (auth, CRUD sensíveis, exportações) enviados a sistema central; playbook de incidente documentado.
- **Responsável / Esforço:** DevOps + Security; **Esforço:** Médio/Alto.

6) Falha de controle de acesso (IDOR / Broken Access Control)

- **Descrição:** Usuário consegue acessar recursos de outro usuário por falta de verificação de autorização (URLs, APIs).
- **Atores:** atacantes legítimos explorando endpoints inseguros.
- **Ativos impactados:** dashboards e relatórios específicos de clientes.
- **Consequências:** exposição de dados de clientes/concorrência, violação contratual.
- **GUT:** G=4, U=4, T=4 → **64**
- **Controles recomendados:** checagem de autorização server-side, uso de claims/tokens com escopo, testes de penetração focados.
- **Critério de aceite:** testes automatizados que tentem acesso horizontal/vertical e falhem; revisão de endpoints.
- **Responsável / Esforço:** Backend; **Esforço:** Médio.

7) Supply-chain: dependências e bibliotecas vulneráveis (npm/pip)

- **Descrição:** Vulnerabilidades em bibliotecas de terceiros comprometem o sistema (ex.: pacote NPM com backdoor).
- **Atores:** atacantes explorando CVEs conhecidos ou atacando manutenção de pacote.
- **Ativos impactados:** aplicação inteira, integridade do ambiente.
- **Consequências:** execução remota, vazamento de dados, compromissos de CI/CD.
- **GUT:** G=4, U=3, T=5 → **60**
- **Controles recomendados:** dependabot / SCA (software composition analysis), política de atualização de dependências, assinatura/verificação de packages.
- **Critério de aceite:** scanner de dependências integrado ao CI; pipeline que bloqueie builds com CVEs críticos.
- **Responsável / Esforço:** Dev + DevOps; **Esforço:** Baixo/Médio.

8) Engenharia social / phishing

- **Descrição:** Usuários (funcionários ou clientes) caem em ataques de phishing e fornecem credenciais.
- **Atores:** phishers, atacantes externos.
- **Ativos impactados:** credenciais, acesso administrativo.
- **Consequências:** comprometimento de contas, fraude.
- **GUT:** G=3, U=3, T=4 → **36**
- **Controles recomendados:** treinamento de conscientização, e-mails com DMARC/DKIM/SPF, MFA.
- **Critério de aceite:** campanha de conscientização realizada; MFA habilitado para admins.
- **Responsável / Esforço:** PO / TI; **Esforço:** Baixo.

9) Armazenamento inseguro de backups/logs (PII em claro)

- **Descrição:** Backups e logs contendo PII armazenados sem criptografia e com acesso amplo.
- **Atores:** quem conseguir acesso ao storage (externo ou interno).
- **Ativos impactados:** backups e logs.
- **Consequências:** vazamento massivo de dados, multas LGPD.
- **GUT:** G=4, U=3, T=3 → **36**
- **Controles recomendados:** criptografia de backups, controle de acesso, ciclo de retenção e destruição segura.
- **Critério de aceite:** backups criptografados; política de retenção definida e aplicada.
- **Responsável / Esforço:** DevOps; **Esforço:** Médio.

10) DDoS / interrupção de disponibilidade

- **Descrição:** Ataque de volumetria interrompe o acesso ao dashboard.
- **Atores:** atacantes de negação de serviço.
- **Ativos impactados:** disponibilidade do serviço.
- **Consequências:** indisponibilidade, perda de receita/uso, frustração do cliente.
- **GUT:** G=4, U=2, T=3 → **24**
- **Controles recomendados:** rate limiting, CDN + WAF, proteção DDoS do provedor (Cloud provider mitigation), escalonamento automático e circuit breakers.
- **Critério de aceite:** políticas de rate limit em endpoints críticos; roteamento via CDN com WAF.
- **Responsável / Esforço:** DevOps; **Esforço:** Médio.

5. Plano de mitigação prioritário

Autenticação segura (Risco 1)

- Implementar MFA para contas administrativas (e recomendável para clientes).
- Forçar políticas de senha e bloquear tentativas repetidas.
- Documentar a arquitetura de autenticação no repositório.

Criptografia e conexão segura (Risco 2)

- Garantir HTTPS em todas as rotas (HSTS quando aplicável).
- Comprovar em ambiente de testes que TLS está ativo.

Remover segredos do repo e ajustar permissões (Risco 3)

- Mover segredos para um secrets manager (ex.: AWS Secrets Manager, Vault).
- Rodar scan para segredos no histórico (ex.: git-secrets) e apresentar evidência.

Sanitização / Prepared Statements (Risco 4)

- Implementar consultas parametrizadas e validar input no backend.
- Adicionar testes automatizados simples que injetem payloads e verifiquem sanidade.

Logging mínimo & playbook (Risco 5)

- Centralizar logs essenciais (auth, exports, alteracões de roles) em um local (pode ser simples para entrega: arquivo centralizado/endpoint mock).
- Entregar um playbook mínimo de 3 passos para incidente (detectar, isolar, comunicar).

Checklist de evidências

- Documento de riscos (este).
- Print/trecho de configuração mostrando HTTPS ativado.
- README com descrição da autenticação/MFA (ou indicação de ferramenta usada).
- Confirmação (print/log) que segredos não estão no repo.
- Teste unitário mostrando input invalidado ou prepared statement.

6. Controles técnicos / configuração recomendada

Autenticação & Autorização: OAuth2 / JWT com refresh token + MFA para admin; RBAC.

Criptografia: HTTPS/TLS; AES-256 at-rest; KMS para chaves.

Input Handling: validação server-side, prepared statements, escape output (XSS).

Headers de segurança: Content-Security-Policy, X-Frame-Options, X-Content-Type-Options, Strict-Transport-Security, Set-Cookie Secure & SameSite.

CORS: permitir apenas origens necessárias (não usar *).

Segurança de dependências: SCA automático no CI; atualizações programadas.

CI/CD: não exibir logs com segredos; pipeline que rode SAST/DAST; reviewers obrigatórios.

Infra & Ops: políticas IAM com menor privilégio; buckets privados; rotação de chaves; backups criptografados.

Monitoramento: alertas auth, uso anômalo de API; logs retidos mínimo X dias (definir política).

Testes e auditoria: integrar SAST e DAST, pen-test antes da entrega final.

7. LGPD — pontos relevantes

- Classificar dados coletados (que são PII).
- Validar bases legais (consentimento, execução de contrato, legítimo interesse).
- Implementar políticas de retenção e exclusão (direito ao esquecimento).
- Garantir mecanismos para atendimento a solicitações de titulares (acesso, correção, exclusão).
- Criptografar dados sensíveis e registrar tratamento de dados (registro de operações).

8. Plano de resposta a incidentes

Detectar — logs e alertas notificam incidente.

Conter — isolar serviço/credenciais afetadas; bloquear IPs/rotas.

Erradicar — remover vetores (corrigir código, revogar chaves).

Recuperar — restaurar de backups limpos; validar integridade.

Lições aprendidas — relatório, comunicação (se necessário) e ajustes de controles.

Responsáveis sugeridos: Product Owner / Time Backend / DevOps / Representante Jurídico (LGPD) / Comunicação.

9. Checklist final

- [ ] Documento de levantamento de riscos e tabela GUT .
- [ ] Plano de mitigação prioritário (itens 1–6 da seção 5).
- [ ] Evidência de HTTPS ativo (print ou resultado de curl).
- [ ] README com instruções de configuração (sem segredos) e arquitetura de autenticação.
- [ ] Exemplo de política de retenção e tratamento de dados (esboço).
- [ ] Pequeno playbook de incidente e lista de contatos internos.
- [ ] (Opcional) Relatório de scan de segredos e dependências (ferramenta CI).

10. Próximos passos recomendados

1. Integrar SCA e SAST no pipeline (CI).
2. Realizar DAST em ambiente de staging.
3. Planejar e executar um pentest (terceiro) antes da entrega final.
4. Estabelecer rotina de backup e teste de restore.
5. Aplicar treinamento básico de segurança para o time e campanhas anti-phishing.

11. Observações finais

Este levantamento foi feito considerando o escopo do Dashboard (admin/cliente, integração com APIs e uso em nuvem). Para aumentar a qualidade das recomendações, no próximo ciclo é interessante ter: (a) diagrama de fluxo de dados (DFD) entre componentes; (b) lista de campos sensíveis reais; (c) detalhes de infraestrutura

(provedor de nuvem, CI/CD). Com essas informações conseguimos calibrar a criticidade e sugerir controles mais específicos (ex.: políticas IAM por recurso, regras WAF precisas).