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 apresenta um levantamento detalhado dos riscos, vulnerabilidades e ameaças associados ao desenvolvimento do **Dashboard Interativo da Cannoli**. O objetivo é fornecer uma visão estruturada dos pontos críticos de segurança, alinhada às etapas do **framework NIST (Identify, Protect, Detect, Respond, Recover)**, a fim de orientar ações preventivas e corretivas para aumentar a resiliência do sistema.

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:** Risco de invasão por credenciais fracas, roubadas ou ausência de autenticação multifator.
- **Etapas NIST:**
- **Identify:** identificar contas críticas e permissões elevadas;
- **Protect:** aplicar MFA e políticas de senha forte;
- **Detect:** monitorar tentativas de login suspeitas;
- **Respond:** redefinir credenciais comprometidas;
- **Recover:** restaurar acesso seguro e revisar políticas.

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

- **Descrição:** Dados pessoais ou estratégicos podem ser interceptados ou acessados sem autorização.
- **Etapas NIST:**

- **Identify:** mapear onde dados sensíveis são armazenados;
- **Protect:** usar criptografia em trânsito e repouso;
- **Detect:** auditar acessos e transferências de dados;
- **Respond:** isolar fontes de vazamento e revogar chaves;
- **Recover:** restabelecer backups seguros e notificar incidentes conforme LGPD.

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

- **Descrição:** Permissões amplas, buckets públicos e chaves expostas em código-fonte.
- **Etapas NIST:**
- **Identify:** localizar serviços e credenciais expostas;
- **Protect:** aplicar políticas IAM e ocultar variáveis sensíveis;
- **Detect:** auditar acessos e monitorar configurações públicas;
- **Respond:** revogar chaves comprometidas;
- **Recover:** validar políticas e realizar testes periódicos de exposição.

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

- **Descrição:** Falhas na validação de entradas permitem execução de comandos maliciosos.
- **Etapas NIST:**
- **Identify:** mapear campos de entrada vulneráveis;
- **Protect:** utilizar queries parametrizadas e validação server-side;
- **Detect:** usar ferramentas SAST/DAST;
- **Respond:** corrigir códigos e restaurar dados afetados;
- **Recover:** implementar testes automatizados para evitar recorrência.

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

- **Descrição:** Ausência de logs relevantes dificulta a identificação e resposta a incidentes.
- **Etapas NIST:**
- **Identify:** definir eventos críticos a serem monitorados;
- **Protect:** centralizar logs e restringir acesso;
- **Detect:** configurar alertas automáticos;
- **Respond:** acionar playbook de resposta a incidentes;
- **Recover:** revisar logs e melhorar processos de detecção.

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

- **Descrição:** Usuário acessa dados ou relatórios de outro cliente por ausência de verificação adequada.
- **Etapas NIST:**
- **Identify:** identificar endpoints que exigem autorização;
- **Protect:** implementar validação server-side;
- **Detect:** realizar testes de penetração simulando acessos indevidos;
- **Respond:** corrigir permissões e alertar usuários impactados;
- **Recover:** validar segurança após correção.

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

- **Descrição:** Bibliotecas externas comprometidas podem introduzir falhas no sistema.
- **Etapas NIST:**
- **Identify:** inventariar dependências usadas;
- **Protect:** aplicar verificações automáticas (SCA, Dependabot);
- **Detect:** monitorar CVEs e pacotes suspeitos;
- **Respond:** remover versões comprometidas;
- **Recover:** atualizar dependências seguras e documentar lições aprendidas.

8) Engenharia social / phishing

- **Descrição:** Usuários ou colaboradores podem ser enganados para fornecer credenciais.
- **Etapas NIST:**
- **Identify:** reconhecer pontos de vulnerabilidade humana;
- **Protect:** realizar treinamentos de conscientização;
- **Detect:** implementar filtros de e-mail e alertas;
- **Respond:** redefinir senhas e revisar políticas;
- **Recover:** reforçar treinamentos e lições aprendidas.

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

- **Descrição:** Dados sensíveis armazenados sem criptografia adequada.
- **Etapas NIST:**
- **Identify:** identificar locais de armazenamento;
- **Protect:** criptografar backups e aplicar controle de acesso;
- **Detect:** verificar integridade dos arquivos;
- **Respond:** isolar e corrigir exposições;
- **Recover:** restaurar dados a partir de cópias seguras.

10) DDoS / interrupção de disponibilidade

- **Descrição:** Alto volume de tráfego pode tornar o dashboard inacessível.
- **Etapas NIST:**
- **Identify:** mapear recursos críticos e pontos de vulnerabilidade;
- **Protect:** implementar rate limiting e uso de WAF/CDN;
- **Detect:** monitorar picos de tráfego e alertas;
- **Respond:** bloquear IPs e redirecionar tráfego;
- **Recover:** restabelecer a disponibilidade e ajustar defesas.

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).