# Vídeo 4 – DevOps com Segurança

## Parte 1 – Problema: Segurança tratada no fim

Pontos-chave:

* - Vulnerabilidades detectadas apenas em produção.
* - Correções caras e impacto ao cliente.
* - Reação lenta a ameaças.

Diagrama: `diagramas/v4\_p1\_seg\_no\_fim.puml`

### Texto de fala:

Quando a segurança vira checklist no final, bugs críticos escapam. Corrigir em produção custa caro e pode exigir rollback amplo.

O cliente vira seu 'tester de segurança', o que é inaceitável. Precisamos antecipar controles e automatizar validações.

## Parte 2 – Discussão: Riscos dessa abordagem

Pontos-chave:

* - Exposição de dados sensíveis e multas.
* - Danos à reputação e perda de confiança.
* - Janelas longas de vulnerabilidade.

Diagrama: `diagramas/v4\_p2\_riscos.puml`

### Texto de fala:

Incidentes geram perdas financeiras, processos regulatórios e danos de imagem. A janela entre descoberta e correção vira um vetor de ataque.

Sem telemetria e políticas, o time opera às cegas e reage tarde. É preciso redesenhar o fluxo com segurança por padrão.

## Parte 3 – Solução: DevSecOps em todo o ciclo

Pontos-chave:

* - Segurança incorporada a cada estágio do DevOps.
* - Responsabilidade distribuída e automatização.
* - Controles como código (policy‑as‑code).

Diagrama: `diagramas/v4\_p3\_devsecops\_cultura.puml`

### Texto de fala:

DevSecOps integra segurança desde o planejamento até a operação, com responsabilidade compartilhada entre Dev, Ops e Sec.

Automatizamos scanners, políticas e validações para que segurança escale com a cadência de releases.

## Parte 4 – Shifting Security Left

Pontos-chave:

* - Detectar cedo é mais barato e seguro.
* - Scanners de SAST/DAST/Dependency early.
* - Educação contínua do time.

Diagrama: `diagramas/v4\_p4\_shift\_left.puml`

### Texto de fala:

Ao trazer segurança para o início, reduzimos o custo de correção e evitamos brechas em produção.

Práticas educativas e ferramentas integradas ao IDE e ao pipeline tornam seguro o caminho padrão.

## Parte 5 – Ferramentas e práticas

Pontos-chave:

* - SAST/DAST/IAST, SCA (dependências).
* - IAM, segredos, hardening e MFA.
* - Monitoramento de ameaças e resposta.

Diagrama: `diagramas/v4\_p5\_ferramentas.puml`

### Texto de fala:

SAST/DAST/IAST e SCA automatizam detecção de vulnerabilidades no código e nas bibliotecas. Gestão de segredos e IAM aplicam o princípio do menor privilégio.

Com hardening, MFA e monitoração contínua, criamos camadas de defesa e reduzimos o impacto de incidentes.

## Parte 6 – Exemplos reais e benefícios

Pontos-chave:

* - Queda de incidentes e MTTR menor.
* - Aumento de confiança do cliente.
* - Compliance mais simples e auditável.

Diagrama: `diagramas/v4\_p6\_beneficios.puml`

### Texto de fala:

Times que adotam DevSecOps reduzem incidentes e respondem mais rápido, com menos impacto ao negócio.

Automação gera trilhas de auditoria e simplifica compliance, fortalecendo reputação e eficiência.