

Briefing Técnico DevOps

Baseado em: Leite et al. (2019): A Survey of DevOps Concepts and Challenges
Levi Dornelas | Matrícula: 01707786 | 5º Período DevOps: UNINASSAU | Fev/2026

Referência do Artigo	Leite et al., ACM Comput. Surv. 52(6), Nov. 2019
Seções Analisadas	1 (Introdução), 4 (Fontes de Conhecimento), 5.2 (People), 6 (Toolset)
Destinatário	Prof. Marcos André: UNINASSAU

1. O que é DevOps: Padronizando a Linguagem

O artigo define DevOps como **"um esforço colaborativo e multidisciplinar dentro de uma organização para automatizar a entrega contínua de novas versões de software, garantindo sua correção e confiabilidade"**. Esta é a definição que adotaremos como linguagem padrão no time.

DevOps surgiu em 2008 como evolução do movimento ágil, preenchendo a lacuna entre desenvolvimento e operações: dois departamentos que historicamente operavam em silos com objetivos conflitantes: dev buscava velocidade de entrega, ops buscava estabilidade. A mediação era um sistema de tickets, lento e propenso a erros.

■■ **Ponto de atenção:** O setor também usa "DevOps" como *job title*, o que gera ambiguidade. Precisamos distinguir internamente: **DevOps como prática organizacional** (o que o artigo define) versus **DevOps Engineer** como cargo técnico.

2. Maturidade da Literatura: O que Ler e o que Ignorar

O survey analisou 167 artigos científicos (50 como core papers) e 238 livros publicados até 2018. As fontes mais relevantes para o time, em ordem de prioridade prática, são:

Fonte	Relevância	Aplicação na Rands
State of DevOps Reports (Puppet/DORA)	Alta: dados quantitativos reais	Benchmarking e justificativa executiva
Accelerate (Forsgren, Humble, Kim)	Alta: ciência de performance	Métricas DORA: deployment freq, MTTR
The Phoenix Project (Gene Kim)	Média: narrativa introdutória	Onboarding cultural de times novos
The DevOps Handbook	Alta: guia prático	Referência operacional do time
IEEE Software Magazine	Média: relatos de campo	Acompanhar tendências

■■ **Risco:** A maioria dos estudos acadêmicos pré-2015 tem baixa qualidade metodológica (confirmado pelo próprio artigo). Não usar estudos antigos como referência para decisões arquiteturais sem verificar se há versão mais recente.

3. A Dimensão Humana: O Desafio Real da Adoção

A seção 5.2 do artigo trata da categoria **People**: é a mais crítica e menos consolidada de todo o survey. Enquanto automação técnica tem boas práticas estabelecidas, as questões organizacionais permanecem sem resposta definitiva. O artigo apresenta três modelos de adoção:

Modelo	Descrição	Risco
--------	-----------	-------

Departamentos colaborantes	Dev e Ops mantêm silos, mas aumentam colaboração	Responsabilidades sobrepostas geram atrito
Times cross-funcionais	"You build it, you run it": time é responsável por dev e ops	Exige profissionais T-shaped raros; suporte 24/7 pesa sobre devs
Time DevOps (bridge team)	Time separado atua como ponte entre dev e ops	Risco de criar um terceiro silo: anti-padrão segundo Humble

■■ **Crítica e risco cultural (contexto Rands):** O artigo não define qual modelo é superior: isso depende do contexto organizacional. Em empresas com cultura hierárquica forte (comum no setor público e grandes corporações brasileiras), forçar times cross-funcionais sem preparação cultural adequada pode resultar em resistência passiva. Recomenda-se mapear o perfil atual antes de escolher o modelo.

■■ **Impacto da LGPD:** O artigo menciona que alguns praticantes defendem acesso irrestrito de engenheiros a dados de produção para diagnóstico. Isso é **incompatível com a LGPD** (Lei 13.709/2018) em sistemas que tratam dados pessoais sensíveis. A Rands deve definir explicitamente quais dados de produção podem ser acessados por engenheiros e com qual processo de auditoria.

4. Mapeamento de Ferramentas: Decisões Técnicas Informadas

A seção 6 do artigo organiza as ferramentas DevOps em seis categorias funcionais, associando-as a conceitos e atores. Esta estrutura é útil para evitar o erro de **tooling-first**: adotar ferramentas sem alinhamento conceitual. O artigo é explícito: estruturar times é mais crítico do que escolher ferramentas específicas.

Categoria	Exemplos	Objetivo DevOps	Risco de Automação
Knowledge Sharing	GitLab Wiki, Rocket Chat, Trello	Colaboração humana	Baixo: mas frequentemente negligenciado
Source Code Mgmt	Git, GitLab, GitHub	Versionamento e colaboração	Baixo: padrão maduro
Build Process	Maven, Gradle, JUnit, SonarQube	Entrega contínua / correção	Médio: testes frágeis (flaky tests) bloqueiam pipeline
CI/CD	Jenkins, GitLab CI, Travis	Pipeline de deploy	Alto: indisponibilidade do CI paralisa entregas
Deployment Automation	Docker, Kubernetes, Chef, Puppet, Heroku	Frequência e confiabilidade	Alto: rollback em BD com LGPD é complexo
Monitoring & Logging	Nagios, Prometheus, Graylog	Confiabilidade em runtime	Médio: alertas excessivos causam fadiga

■■ **Controvérsia identificada no artigo: Containers vs. Configuration Management:** O artigo aponta debate não resolvido entre abordagem containerizada (Docker/Kubernetes) e gerenciamento de configuração contínuo (Chef/Puppet). A tendência da comunidade é favorecer containers, mas há casos legítimos de uso complementar. A Rands deve definir sua posição antes de iniciar novos projetos para evitar ambientes híbridos não gerenciáveis.

5. Recomendações para o Time de Arquitetura

#	Recomendação	Prioridade
1	Adotar a definição do artigo como linguagem oficial de DevOps na Rands	Alta

2	Usar as 4 métricas DORA (lead time, deploy frequency, MTTR, change failure rate) como baseline de avaliação	Alta
3	Realizar mapeamento do modelo organizacional atual antes de definir estrutura de times	Alta
4	Criar política explícita de acesso a dados de produção alinhada à LGPD	Alta
5	Decidir posicionamento oficial: containers vs. config management: evitar hibridismo sem governança	Média
6	Não iniciar com todas as ferramentas: começar pelo pipeline CI/CD e expandir gradualmente	Média
7	Incluir investimento em ferramentas de knowledge sharing: área frequentemente negligenciada	Baixa

Briefing elaborado com base nas Seções 1, 4, 5.2 e 6 de: Leite, L. et al. *A Survey of DevOps Concepts and Challenges*. ACM Computing Surveys, Vol. 52, No. 6, Art. 127, Nov. 2019. | Entregue em: 20/02/2026