



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Modelo de Avaliação DevOps em Ambientes de Nuvem com Métricas DORA: Um Estudo no Setor Financeiro

Emanuel Galhardo de Oliveira

Dissertação apresentada como requisito para defesa final do
Mestrado Profissional em Computação Aplicada

Orientador

Prof. Dr. George Marsicano Corrêa

Brasília
2025

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

0048mm Oliveira, Emanuel Galhardo de
Modelo de Avaliação DevOps em Ambientes de Nuvem com
Métricas DORA: Um Estudo no Setor Financeiro / Emanuel
Galhardo de Oliveira; orientador George Marsicano Corrêa;
co-orientador Edna Dias Canedo. Brasília, 2025.
105 p.

Dissertação(Mestrado Profissional em Computação Aplicada)
Universidade de Brasília, 2025.

1. DevOps. 2. Métricas DORA. 3. Computação em Nuvem. 4.
Setor Financeiro. I. Marsicano Corrêa, George, orient. II.
Dias Canedo, Edna, co-orient. III. Título.

Dedicatória

À presença nos momentos, ao significado das ações, ao aprendizado constante e à alegria sincera de buscar oferecer o melhor de si. Dedico este trabalho à Aninha e ao Samuel.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

A adoção de práticas DevOps tornou-se um elemento central para organizações que buscam reduzir fricções entre desenvolvimento e operações, acelerar a entrega de software e aumentar a confiabilidade dos serviços em ambientes distribuídos e baseados em nuvem. Apesar da ampla difusão dessas práticas, ainda persiste a ausência de um modelo sistemático que permita avaliar o nível de evolução dos times a partir de indicadores mensuráveis, confiáveis e comparáveis. Este estudo enfrenta esse desafio ao propor um referencial de avaliação fundamentado nas métricas DORA (**DevOps Research and Assessments**), integrando evidências técnicas, operacionais e sociotécnicas. A pesquisa utiliza um estudo de caso em uma grande instituição financeira, combinando extração automatizada de indicadores, análise documental, entrevistas semiestruturadas e triangulação qualitativa para interpretar percepções, práticas e limitações estruturais dos times. Os resultados mostram padrões consistentes entre os valores observados das métricas e as percepções dos grupos sobre restrições, gargalos e capacidades operacionais, evidenciando convergências importantes entre dados técnicos e fatores humanos e organizacionais. Conclui-se que a aplicação integrada das métricas DORA permite caracterizar de forma objetiva o estágio de desenvolvimento das práticas DevOps, oferecendo insumos para decisões de melhoria contínua e estabelecimento de processos sustentáveis, ainda que limitada por aspectos como amplitude tecnológica, dependência de APIs e necessidade de maior padronização.

Palavras-chave: *DevOps*, Métricas DORA, Computação em Nuvem, Setor Financeiro.

Abstract

The adoption of DevOps practices has become a central element for organizations seeking to reduce friction between development and operations, accelerate software delivery, and increase service reliability in distributed, cloud-based environments. Despite the widespread diffusion of these practices, there remains a lack of a systematic model capable of assessing the evolution of teams through measurable, reliable, and comparable indicators. This study addresses this challenge by proposing an evaluation framework grounded in DORA (**DevOps Research and Assessments**) metrics, integrating technical, operational, and sociotechnical evidence. The research employs a case study in a large financial institution, combining automated extraction of operational indicators, document analysis, semi-structured interviews, and qualitative triangulation to interpret perceptions, practices, and structural limitations of the teams. The results reveal consistent patterns between the observed metric values and team perceptions regarding constraints, bottlenecks, and operational capabilities, highlighting significant convergences between technical data and human and organizational factors. The study concludes that the integrated application of DORA metrics enables an objective characterization of the developmental stage of DevOps practices, providing inputs for continuous improvement and the establishment of sustainable processes, while acknowledging limitations such as technological scope, dependence on APIs, and the need for greater process standardization.

Keywords: DevOps, DORA Metrics, Cloud Computing, Financial Sector.

Sumário

1	Introdução	1
1.1	<i>DevOps</i> no Setor Financeiro	3
1.2	Problema	4
1.3	Justificativa	5
1.4	Motivação	6
1.5	Objetivos	8
1.5.1	Objetivo Geral	8
1.5.2	Objetivos Específicos	9
1.6	Estrutura do Trabalho	9
2	Fundamentação Teórica	10
2.1	<i>DevOps</i>	10
2.1.1	<i>Pipeline</i> em <i>DevOps</i>	13
2.2	Frameworks de Avaliação <i>DevOps</i>	16
2.3	Métricas DORA	17
2.3.1	Métricas de Velocidade	19
2.3.2	Métricas de Estabilidade	20
2.3.3	Desempenho Operacional	20
2.4	Sobre a Instituição Financeira	22
2.5	Trabalhos Correlatos	24
3	Metodologia	35
3.1	Procedimento de Revisão da Literatura	37
3.1.1	Busca Inicial <i>Ad Hoc</i>	37
3.1.2	Busca Complementar por <i>Snowballing</i>	38
3.2	Análise Documental	39
3.3	Lógica de Primeira Ordem	40
3.4	Revisão de Ferramentas	41
3.5	Entrevista Aberta	41

3.5.1	Entrevista Exploratória	42
3.5.2	Grupo Focal	44
3.6	Análise de Requisitos	46
3.7	Modelagem Conceitual	47
3.7.1	Linguagem de Modelagem Unificada (UML)	48
3.8	Prova de Conceito	50
3.8.1	Arquitetura <i>Model-View-Controller</i> (MVC)	50
3.8.2	Programação Orientada a Objetos	52
3.9	Estudo de Caso	54
3.9.1	Componentes do Estudo	55
3.9.2	Tipos de Serviços Sustentados pelos Times <i>DevOps</i>	57
4	Resultados	58
4.1	Estratégias de Pesquisa	58
4.1.1	Evidências da Busca Inicial <i>Ad Hoc</i>	59
4.1.2	Evidências da Busca Complementar por <i>Snowballing</i>	59
4.2	Algoritmos das Métricas	60
4.3	Funcionamento e Restrições do Modelo	62
4.3.1	Ferramentas Candidatas	62
4.3.2	Extração das Métricas de Velocidade	63
4.3.3	Extração das Métricas de Estabilidade	65
4.3.4	Avaliação do Modelo com Especialistas	65
4.3.5	Validação Qualitativa e Triangulação dos Dados	69
4.3.6	Restrições do Modelo	70
4.3.7	Considerações sobre APIs e Extração de Logs	71
4.4	Levantamento de Requisitos	72
4.4.1	Requisitos Funcionais	72
4.4.2	Requisitos Não Funcionais	73
4.5	Arquitetura da Solução	73
4.6	Código-fonte das APIs	80
4.7	Avaliação do Desempenho <i>DevOps</i>	80
4.7.1	Componentes do Estudo	81
4.7.2	<i>DORA Quick Check</i>	83
4.7.3	Avaliação das APIs	84
4.7.4	Investigação com Grupos Focais	85
4.7.5	Percepções sobre o Uso das Métricas DORA	86
4.7.6	Integração dos Resultados Quantitativos e Qualitativos	88
4.7.7	Síntese das Percepções dos Grupos Focais	89

4.7.8	Comparação do Uso das Métricas DORA	89
5	Discussão de Resultados	91
6	Considerações Finais	95
6.1	Síntese dos Resultados	95
6.2	Contribuições	96
6.3	Limitações	96
6.4	Conclusão	97
6.5	Trabalhos Futuros	97

Lista de Figuras

2.1	Ciclo do <i>DevOps</i>	11
2.2	Mapa mental do DevOps: cultura, práticas, processos, automação, observabilidade e operações.	12
2.3	Modelo de Sucesso <i>DevOps</i> na <i>Netflix</i> , integrando cultura, arquitetura, ferramentas e habilidades.	27
3.1	Fluxo da Metodologia de Pesquisa <i>Snowballing</i>	38
3.2	Tipos de diagramas da UML.	48
3.3	Fluxo de interação entre os componentes da arquitetura MVC.	51
3.4	Exemplo de Diagrama de Classes representando Contas Bancárias.	53
4.1	Diagrama de Componentes das Camadas de API e do Consumidor.	75
4.2	Diagrama de Classes da Visão <i>Resource</i>	76
4.3	Diagrama de Classes da Visão Repositório.	77
4.4	Diagrama de Classes da Visão <i>Cluster</i>	79

Lista de Tabelas

1.1	Relação entre problemas identificados no instituição e Métricas DORA. . .	5
2.1	Comparação entre frameworks de avaliação em <i>DevOps</i>	16
2.2	Métricas de Desempenho de Entrega de <i>Software</i>	18
2.3	Avaliação de Desempenho Operacional	21
3.1	Metodologia em Fases	36
3.2	Classificação dos serviços sustentados pelos times <i>DevOps</i>	57
4.1	Lista dos Critérios de Inclusão e Exclusão	58
4.2	Critérios de validação de qualidade do <i>software</i>	64
4.3	Integração qualitativa entre métricas DORA, entrevistas e percepções dos especialistas.	70
4.4	Comparação entre uso de APIs e extração de logs como fonte para mensu- ração das métricas DORA.	71
4.5	Classificação dos requisitos não funcionais segundo URPS+	73
4.6	Lista das Questões de Estudo	81
4.7	Resumo da avaliação de desempenho <i>DevOps</i> pelo DORA Quick Check . .	84
4.8	Situação antes e depois da adoção das métricas DORA	89

Lista de Abreviaturas e Siglas

API *Application Programming Interface.*

B2B *Business-to-Business.*

CAPES Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

CI/CD Integração e entrega contínuas.

DevOps Composto dos perfis de Desenvolvimento (Dev) e Operações (Ops).

DORA *DevOps Research and Assessments.*

SOX *Sarbanes-Oxley Act.*

TI Tecnologia da Informação.

Capítulo 1

Introdução

Embora os negócios entendam que novas tecnologias tragam vantagens em sua atuação e tempestividade nas constantes necessidades estratégicas, viabilizar um novo domínio técnico e fluidez para se trabalhar demanda mudanças culturais (LEE; GREWAL, 2004). Pensando nisso, em 2018, motivado pela percepção do mercado, o Composto dos perfis de Desenvolvimento (Dev) e Operações (Ops) (DevOps) foi priorizado em uma instituição financeira como um assunto de relevância estratégica. Para tanto, foram dedicados profissionais de tecnologia com o desafio de pesquisar, prospectar e avaliar a aplicabilidade do *DevOps* na organização.

Em uma conferência na Bélgica intitulada “*DevOps days*” em 2009 organizada pelo agilista Patrick Debois, o termo *DevOps* foi apresentado pela primeira vez. Trata-se da integração de tarefas de desenvolvimento e operações em um time único (WIEDEMANN et al., 2023). Tradicionalmente, as organizações integram times *DevOps* para apoiar a entrega contínua de *software* (FITZGERALD; STOL, 2017). A empresa de pesquisa de mercado *Gartner* estima que cerca de 90% das organizações utilizam *DevOps* para apoiar a entrega de valor ao cliente (HAIGHT; SPAFFORD, 2022). A partir dessa percepção, a cultura foi expandida para toda a organização. Desde então, uma comunidade interna colabora regularmente compartilhando informações, definindo roteiros, aprimorando habilidades e trabalhando na evolução das práticas adotadas pelos times.

Para trazer a nova forma de trabalho, houve uma mobilização na atualização da infraestrutura, com a virtualização de servidores, redes e armazenamento. Um novo contexto no ambiente de trabalho foi consolidado para estabelecer um ciclo de desenvolvimento padronizado, com automação nos processos de testes, empacotamento, validação e publicação com fases de preparação, piloto e produção das soluções.

Com o novo parque tecnológico e metodologias, exigiu-se adaptação da governança para acompanhamento do uso dos recursos. Para um funcionamento adequado, as auto-

mações operacionais relativas à entrega de código nos repositórios, compilações e publicações nos ambientes devem operar conforme esperado (KIM et al., 2021).

Desde então, dada a amplitude de produtos mobilizados, existem mais de 300 times *DevOps* atuando na realização de entregas. Nesse cenário há dificuldade em manter a estabilidade dos serviços, tais como o provisionamento do ambiente, consistência nos processos operacionais e rastreabilidade de incidentes. Cada time atua gerenciando um conjunto de microsserviços que representam produtos ou uma plataforma. Além da responsabilidade em manter implementações existentes, os times mantêm um *backlog*¹ de funcionalidades e evolução junto aos patrocinadores.

Quanto à manutenção, monitoração e resposta a incidentes em ambientes de produção, esses times enfrentam dificuldades relacionadas ao tempo de atendimento. Além disso, uma vez superados os incidentes, o postmortem — análise da origem do problema específico ocorrido e consolidação das lições aprendidas — nem sempre é realizado (ROSENTHAL; JONES, 2020). A falta desse processo favorece reincidência por falta de investigação da origem do problema.

Incidentes priorizam ações tempestivas que podem gerar risco de imagem e impactos ao cliente. O trabalho não planejado e o retrabalho são indicadores úteis porque representam falhas no processo. O resultado é a ampliação do volume de demandas sob responsabilidade do time (KIM et al., 2021). Com dificuldades de priorização em suas atividades, mudanças críticas, como entregas motivadas por exigências legais, podem ser afetadas.

Além disso, existem limitações na monitoração, acompanhamento e atuação nas soluções. O acesso aos *logs*² e a execução de comandos de operação necessários para superar cenários de indisponibilidade são restritos à gerência responsável pela infraestrutura nuvem. Em vez da atuação direta do time, para intervenção é necessário criar *issues* ou registros de incidentes que entram em uma fila de atendimento.

Um exemplo recorrente é a indisponibilidade da esteira de *deploy*³ para publicação de microsserviços no ambiente de produção. Esse tipo de indisponibilidade afeta o fluxo de entregas contínuas (FORSGREN; HUMBLE; KIM, 2018).

De um lado existe a autonomia pretendida dos times para atuarem com seus produtos; de outro, existem restrições necessárias por motivos de estabilidade. Para ilustrar essa criticidade, existem soluções que representam o core bancário dos serviços financeiros, isto é, funcionalidades cujo processamento não pode ser interrompido devido ao impacto econômico potencial. Funcionalidades tais como pagamentos, transferências, depósitos

¹Histórias de usuário registradas pelo dono do produto com futuro potencial de atendimento pelo time.

²Mensagens armazenadas para monitorar e compreender a operação do sistema, possibilitando depurar problemas ou auditar funcionalidades.

³Envio de mudanças ou atualizações de uma solução para um ambiente de implantação.

e empréstimos demandam disponibilidade contínua (HARALAYYA, 2021). Além disso, há exigências legais definidas por órgãos reguladores, como a *Sarbanes-Oxley Act* (SOX), incluindo a possibilidade de multas caso soluções estejam inoperantes (STATES, 2002).

Em síntese, este trabalho apresenta uma contribuição científica estruturada em três dimensões complementares: (i) a formalização de um **modelo** de mensuração do desempenho *DevOps* baseado nas métricas DORA, adaptado ao contexto regulado de instituições financeiras de grande porte; (ii) o desenvolvimento de um **método** técnico-metodológico inédito, que combina análise documental, lógica de primeira ordem, integração de dados via APIs e entrevistas qualitativas; e (iii) a realização de uma **análise inédita** sobre a aplicação das métricas DORA em ambientes financeiros regulados, destacando barreiras culturais, organizacionais e técnicas. Essas três dimensões consolidam a originalidade da pesquisa e estabelecem sua relevância acadêmica e prática, oferecendo fundamentos para futuras investigações e aplicações em diferentes contextos do setor financeiro.

1.1 *DevOps* no Setor Financeiro

O setor financeiro ocupa posição estratégica na economia global, caracterizado por operações de alta criticidade, rigor regulatório e necessidade de disponibilidade contínua dos serviços. Bancos e instituições financeiras lidam diariamente com milhões de transações que envolvem pagamentos, transferências, investimentos e crédito, em ambientes que não podem sofrer interrupções sem gerar impactos econômicos e sociais significativos. Nesse contexto, a adoção de práticas modernas de engenharia de software torna-se essencial para garantir agilidade e confiabilidade.

A transformação digital ampliou a pressão sobre as instituições financeiras, que passaram a competir não apenas entre si, mas também com *fintechs* e empresas de tecnologia que oferecem soluções inovadoras e altamente escaláveis. Essa competição exige que bancos tradicionais adotem metodologias capazes de reduzir o tempo de entrega de novas funcionalidades, aumentar a qualidade das soluções e assegurar conformidade regulatória. O *DevOps* surge como resposta a esse cenário, integrando desenvolvimento e operações para acelerar ciclos de entrega e reduzir falhas.

Entretanto, a aplicação do *DevOps* em instituições financeiras deve considerar a complexidade dos sistemas legados, a diversidade de plataformas, a necessidade de integração com serviços externos e a exigência de auditoria e rastreabilidade tornam a implementação mais difícil do que em setores menos regulados. Além disso, a escala das operações e a heterogeneidade dos times aumentam a necessidade de métricas objetivas que permitam avaliar desempenho e evolução das práticas.

As métricas *DevOps Research and Assessments* (DORA) constituem uma abordagem baseada em evidências para mensurar aspectos críticos da entrega contínua (frequência de implantações, tempo para mudanças, taxa de falhas e tempo de recuperação). No setor financeiro, essas métricas possibilitam tanto o monitoramento técnico quanto a avaliação da atuação dos times em ambientes de nuvem, apoiando decisões estratégicas voltadas à melhoria contínua e à conformidade regulatória.

Assim, compreender o papel do *DevOps* no setor financeiro é fundamental para contextualizar o estudo de caso realizado nesta instituição. Os desafios enfrentados pela instituição refletem problemas comuns a organizações financeiras de grande porte, e a proposta de um modelo de avaliação baseado em métricas DORA busca oferecer contribuições que extrapolam o caso específico, podendo ser aplicadas em diferentes contextos do setor.

1.2 Problema

Diante do contexto apresentado, a instituição financeira foco deste estudo enfrenta desafios relacionados à gestão de vários times *DevOps*, à qualidade dos serviços, ao tempo de resposta e rastreabilidade dos incidentes, à priorização das atividades, ao monitoramento e intervenção nas soluções devido às restrições definidas pela gerência de infraestrutura nuvem e ao equilíbrio entre autonomia dos times e estabilidade das soluções.

Tal conjunto de itens relacionados à implementação da cultura *DevOps* reflete problemas recorrentes em organizações que buscam transformação digital (KIM et al., 2021). Como alternativa, as métricas DORA permitem explicitar o nível de desempenho dos times *DevOps*. Essas métricas apoiam o monitoramento contínuo, abrangendo frequência de implantações, tempo para mudanças, tempo de recuperação, taxa de falhas e comportamento operacional (FORSGREN; HUMBLE; KIM, 2018). Os detalhes sobre seu funcionamento são apresentados no capítulo 2.

Ao analisar as métricas DORA em termos de seus níveis e métodos de aquisição de desempenho em times *DevOps*, foi possível identificar métricas com potencial para apoiar a instituição no enfrentamento dos desafios mencionados. Com monitoramento contínuo, é possível identificar oportunidades de melhoria e implementar ações corretivas para aprimorar os processos (FORSGREN; HUMBLE; KIM, 2018).

A tabela 1.1 indica quais métricas DORA podem ser úteis para monitorar problemas específicos.

Problemas na instituição	Métrica DORA recomendada
Gestão de vários times <i>DevOps</i>	Desempenho operacional
Garantia da qualidade dos serviços	Tempo para mudanças
Tempo de resposta aos incidentes	Tempo de recuperação e taxa de falhas
Rastreabilidade em incidentes	Taxa de falhas
Monitoramento e intervenção nas soluções	Frequências de entregas
Autonomia dos times	Desempenho operacional
Estabilidade das soluções	Tempo para mudanças

Tabela 1.1: Relação entre problemas identificados no instituição e Métricas DORA.

Diante dessas oportunidades, a adoção de práticas *DevOps* sustentadas por métricas DORA pode apoiar a instituição com instrumentos para lidar com os desafios mencionados. No entanto, o Banco ainda carece de uma estratégia definida para coletar, analisar e aplicar tais métricas.

1.3 Justificativa

Times *DevOps* devem ser capazes de criar, testar e implantar código de forma contínua, garantindo conformidade e a estabilidade das entregas (BIRD, 2015). O uso de tecnologias sem o devido alinhamento com a cultura corporativa limita a adoção plena dessas práticas. A efetividade depende da comunicação e da definição de metas compartilhadas entre os membros dos times, incluindo desenvolvedores, analistas e gestores de valor da solução (KIM et al., 2021).

A entrega contínua visa reduzir a latência no desenvolvimento das soluções. Isso implica reduzir o tempo de disponibilização de versões, automatizando etapas de construção, testes e publicação. A automação se torna viável quando times possuem processos consistentes de integração e modernização de aplicações (KIM et al., 2021).

Em modelos organizacionais tradicionais, desenvolvedores atuam com pouca coordenação com os times de operações responsáveis pelos ambientes de produção. Após o desenvolvimento, as entregas são encaminhadas para publicação. Caso a solução não funcione adequadamente, os times atuam separadamente, o que gera atrasos e fragmentação no atendimento (ROSENTHAL; JONES, 2020).

Em ambientes *DevOps*, desenvolvimento e operação atuam com alinhamento, reduzindo o tempo de mobilização para todo o processo. Quando o código-fonte⁴ é implantado com sucesso, o time responsável acompanha o comportamento da solução em produção.

⁴Conjunto de instruções escritas por um programador usando uma linguagem de programação.

Em caso de incidentes, há capacidade de atuação mais imediata (FORSGREN; HUMBLE; KIM, 2018).

A adoção contínua dessas práticas requer que as organizações definam como áreas distintas irão trabalhar em conjunto. Isso implica ajustar responsabilidades, revisar processos e capacitar times para suportar a operação e evolução das soluções (KIM et al., 2021). A aprendizagem contínua e a transferência de conhecimento se tornam essenciais para consolidar esses comportamentos a longo prazo.

O estabelecimento de um modelo de métricas *DevOps* é necessário para ampliar a autonomia dos times. O uso de métricas permite monitorar o comportamento operacional, apoiar decisões e orientar ações de melhoria. Com isso, os times podem identificar limitações, direcionar ajustes nos processos e evoluir de maneira sistemática ao longo do tempo.

1.4 Motivação

A ampliação das práticas de *DevOps* decorre da necessidade de estabelecer mecanismos que permitam lidar com o aumento da complexidade técnica, com o volume de demandas e com a interdependência crescente entre produtos e plataformas. A adoção de serviços em nuvem, pipelines automatizados e práticas de integração contínua exige que os times atuem com processos consistentes, reduzindo a variabilidade do fluxo de desenvolvimento e aumentando a previsibilidade operacional. No entanto, mesmo com a disseminação de práticas automatizadas, ainda há limitações na visibilidade sobre o comportamento das soluções, o que dificulta a identificação de padrões, desvios e fatores estruturais que influenciam o desempenho.

As práticas de *DevOps* têm como propósito integrar desenvolvimento e operações, reduzindo o intervalo entre a criação de uma mudança e sua disponibilização em produção (FORSGREN; HUMBLE; KIM, 2018). Contudo, a adoção dessas práticas de forma isolada não garante que os times consigam mensurar o impacto efetivo de suas ações, especialmente em contextos caracterizados por ampla diversidade de produtos e múltiplas dependências internas. No caso de instituições financeiras, tais limitações tornam-se ainda mais evidentes em razão da complexidade das operações, do rigor regulatório e da coexistência de sistemas legados com serviços modernos baseados em nuvem (BIRD, 2015).

A criação de ambientes em nuvem, a virtualização de componentes e a expansão dos serviços digitais aumentaram a demanda por mecanismos formais que permitam analisar o comportamento dos times em todo o ciclo de vida do *software*. O uso de métricas torna-se essencial para apoiar decisões, identificar distorções nos fluxos e compreender como

práticas, processos e estruturas organizacionais influenciam os resultados. As métricas DORA oferecem uma abordagem baseada em evidências para observar dimensões do processo de entrega - frequência de implantações, tempo para mudanças, taxa de falhas e tempo de recuperação - permitindo análises comparáveis entre times, períodos e tipos de serviços (FORSGREN; HUMBLE; KIM, 2018).

A motivação deste trabalho está em estabelecer um modelo que permita compreender de forma sistemática o comportamento operacional dos times *DevOps*. A instituição possui serviços que variam desde *Application Programming Interface* (API)s de alto volume até processos internos com requisitos específicos de conformidade. Essa diversidade aumenta a necessidade de parâmetros objetivos que permitam avaliar o impacto das mudanças nos serviços e o comportamento das soluções em produção. A ausência de tais parâmetros cria dependência de percepções individuais, o que dificulta priorização, planejamento e identificação de limitações estruturais.

Além disso, a atuação em nuvem — dependente de automações, monitoramento distribuído e orquestração de contêineres — demanda métricas que permitam observar o comportamento dos serviços de forma contínua. Práticas como análise de código, testes automatizados, observabilidade e automação de segurança precisam ser acompanhadas por indicadores que evidenciem se essas ações de fato reduzem o tempo de ciclo ou contribuem para a estabilidade dos serviços (KIM et al., 2021). A falta de tais indicadores reduz a capacidade de identificar pontos de atenção, padronizar processos ou orientar intervenções.

A instituição enfrenta também desafios relacionados à gestão de incidentes, à execução de postmortem e à limitação de acesso a ferramentas de operação e monitoramento. Tais fatores geram retrabalho, aumentam o volume de demandas não planejadas e reduzem a capacidade de resposta. A adoção de métricas permite explicitar esses efeitos e medir sua relação com práticas internas, evidenciando padrões que apoiam decisões de priorização.

Assim, a motivação deste trabalho envolve criar um mecanismo que permita aos times observar tendências, comparar períodos e identificar fatores que influenciam a entrega de valor. O uso estruturado das métricas DORA busca preencher lacunas relacionadas ao acompanhamento da operação, oferecendo uma base empírica para evolução incremental e planejamento de melhorias. Com isso, espera-se apoiar a instituição na consolidação de práticas que promovam maior previsibilidade, menor variabilidade no fluxo e maior capacidade de tomada de decisão orientada por dados.

O monitoramento contínuo, aliado à automação de respostas, reduz o tempo de recuperação de incidentes críticos e suporta a disponibilidade dos serviços. A adoção de práticas de containerização e orquestração contribui para a escalabilidade dos ambientes, ajustando recursos conforme a demanda. O uso de ferramentas de análise de código e se-

gurança nos pipelines tem como objetivo diminuir vulnerabilidades e não conformidades após a implantação.

A seguir, estão os principais resultados esperados:

- **Velocidade de Entrega de *Software*:** A eficiência da esteira automatizada é observada no tempo de ciclo entre o desenvolvimento e a implantação em produção, indicando a capacidade de disponibilizar alterações em intervalos menores.
- **Redução de Incidentes e Tempo de Recuperação:** A maturidade dos serviços em nuvem pode ser medida pela variação no número de incidentes de maior impacto e no tempo de recuperação após uma falha, relacionados às práticas de monitoramento e automação de resposta.
- **Escalabilidade e Eficiência Operacional:** A capacidade de ajustar recursos de forma automática, considerando restrições de custo e disponibilidade, representa um indicador associado à maturidade do ambiente em nuvem. Times que operam mecanismos de orquestração e controle de carga tendem a apresentar menor variação no uso de recursos.
- **Qualidade e Segurança do Código:** A utilização de ferramentas automatizadas de análise estática e de segurança no pipeline busca reduzir vulnerabilidades e não conformidades, permitindo que os times monitorem o impacto das práticas de desenvolvimento e verifiquem aderência aos requisitos de segurança.
- **Colaboração e Feedback Contínuo:** O desempenho também pode ser observado pelo nível de integração entre times de desenvolvimento e operações. Ferramentas de feedback contínuo e métricas de satisfação de desenvolvedores permitem avaliar se as práticas adotadas favorecem ciclos de trabalho mais coordenados.

1.5 Objetivos

1.5.1 Objetivo Geral

Propor e validar um modelo de avaliação de desempenho de times *DevOps* em ambientes de computação em nuvem, utilizando as métricas DORA como núcleo de mensuração. O modelo busca caracterizar não apenas o desempenho técnico de entrega e operação, mas também o estágio evolutivo das práticas *DevOps*, integrando dimensões técnicas, operacionais e sociotécnicas.

1.5.2 Objetivos Específicos

- OE1 – Revisar fundamentos conceituais sobre modelos de métricas *DevOps* e o papel das métricas DORA na avaliação de desempenho e evolução das práticas.
- OE2 – Analisar técnicas e práticas aplicáveis para extração das métricas DORA, considerando restrições técnicas e operacionais da instituição financeira em estudo.
- OE3 – Definir pontos de coleta de dados necessários para o cálculo das métricas DORA, avaliando mecanismos disponíveis nos ambientes de nuvem e nas ferramentas de integração e entrega contínua.
- OE4 – Estruturar um modelo preliminar de avaliação *DevOps*, fundamentado nas métricas DORA e nos fluxos operacionais utilizados pelos times.
- OE5 – Implementar uma prova de conceito (POC) com mecanismos de coleta de dados necessários ao cálculo de cada métrica por microsserviço.
- OE6 – Validar o modelo por meio de aplicação prática em times *DevOps*, comparando percepções dos participantes com os valores obtidos das métricas, de modo a verificar a aderência entre indicadores técnicos e fatores organizacionais.

1.6 Estrutura do Trabalho

Este trabalho está organizado em seis capítulos, descritos a seguir:

O Capítulo 2 apresenta a fundamentação teórica e os trabalhos correlatos relacionados à atuação de times *DevOps*.

O Capítulo 3 descreve a metodologia adotada para a recuperação das métricas DORA, considerando as ferramentas disponíveis na instituição em estudo.

O Capítulo 4 apresenta os resultados preliminares da aplicação da metodologia, incluindo a análise da viabilidade de mensuração do desempenho *DevOps* por meio das métricas DORA.

O Capítulo 5 contém a análise dos resultados e a identificação de oportunidades de aprimoramento no processo de mensuração das métricas DORA.

O Capítulo 6 apresenta as considerações finais do trabalho.

Capítulo 2

Fundamentação Teórica

Para contextualizar o assunto específico desta dissertação, é apresentado um conjunto de fundamentos teóricos que direcionam e esclarecem a aplicabilidade das métricas DORA. A expectativa com uso das métricas é permitir um melhor desempenho operacional em times *DevOps* na instituição financeira foco deste estudo.

2.1 *DevOps*

O Composto dos perfis de Desenvolvimento (Dev) e Operações (Ops) (DevOps) não é somente sobre tecnologia. A tecnologia possui limites definidos, enquanto as pessoas, aspectos organizacionais e culturais são evoluídos de maneira ostensiva (HUMBLE; MOLESKY; O'REILLY, 2020). Empresas por vezes ficam concentradas nas ferramentas, e desconsideram o impacto além da tecnologia. Uma etapa necessária na construção do processo consiste em compreender as mudanças culturais e organizacionais, pois elas condicionam o sucesso da adoção do *DevOps*.

A busca por uma forma específica e otimizada de atuação de times dedicados à entrega de *software* pode ser funcional em determinada organização, mas não deve ser entendida como uma receita universal. O *DevOps* é influenciado por um conjunto de fatores culturais, processuais e tecnológicos que condicionam sua aplicação. Ressalta-se que não existe um modelo único de mobilização do *DevOps*; trata-se de uma integração entre pessoas, processos e tecnologias voltada à entrega contínua de valor ao cliente (SONI, 2020).

Diferente do modelo tradicional com atuação isolada e independente, desenvolvedores e operadores, além de outros especialistas, trabalham integrados desde o início do ciclo de vida de desenvolvimento da solução até a implantação, além do acompanhamento em produção. O foco é na aceleração das entregas e obtenção de uma melhor propriedade, e correspondente, uma melhor qualidade do produto final (WINTERS; MANSHRECK; WRIGHT, 2020).

Clientes são atendidos da melhor maneira, com entregas contínuas e uma qualidade aprimorada nos produtos. No entanto, com os benefícios que times *DevOps* oferecem, uma atuação plena é complexa. Seja alinhando as metas e prioridades para promover a colaboração entre times multifuncionais, capacitando tecnicamente seus integrantes, ou definindo estratégias de migração para sistemas legados (FORSGREN; HUMBLE; KIM, 2018).

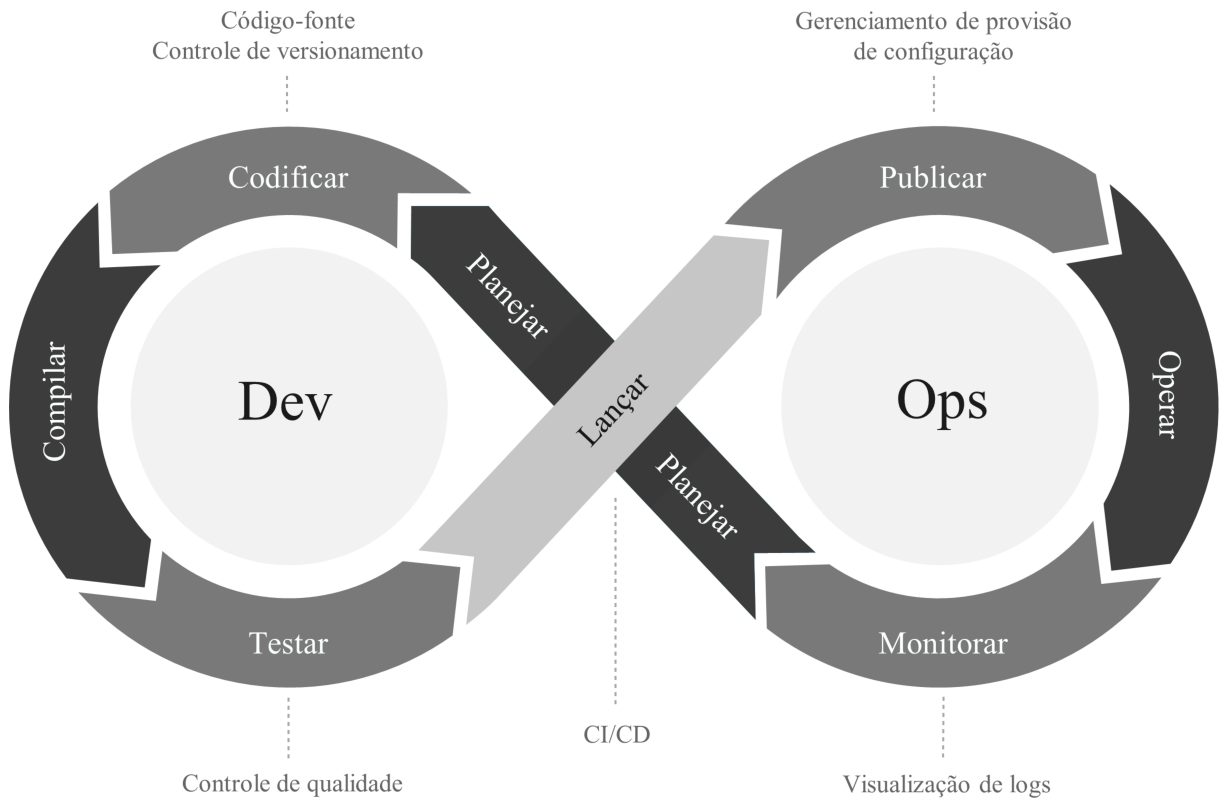


Figura 2.1: Ciclo do *DevOps*.

Existe um ciclo recorrente focado no refinamento e na evolução de uma solução, conforme figura 2.1. Suas práticas culturais, arquitetônicas e técnicas resultantes representam uma convergência de muitos movimentos de maturidade e gerenciamento na indústria da Tecnologia da Informação (TI). Há décadas de lições aprendidas, contribuições de organizações de alta confiabilidade, modelos de gerenciamento de qualidade que definiram a sua atual referência (KIM et al., 2021).

O *DevOps* resulta da aplicação dos princípios mais consolidados do domínio da produção e da liderança ao fluxo de valor de TI. Baseia-se nos conhecimentos de *Lean*⁵, Teoria

⁵Metodologia que busca eliminar desperdícios, em diversos níveis, dentro das empresas.

das Restrições⁶, Sistema Toyota de Produção⁷, Engenharia de Resiliência⁸, Organizações de Aprendizado⁹, Cultura de Segurança¹⁰ e Fatores Humanos¹¹ (HUMBLE; MOLESKY; O'REILLY, 2020).

Outro contexto relevante do qual o *DevOps* se inspira é o gerenciamento de autoconfiança, liderança servidora e gerenciamento de mudanças organizacionais (KIM et al., 2021).

O resultado é qualidade, confiabilidade, estabilidade e segurança de classe mundial com custos e esforços cada vez menores e fluxo e confiabilidade acelerados em todo o fluxo de valor da tecnologia, incluindo gerenciamento de produtos, desenvolvimento, controle de qualidade, operações de TI e segurança da informação (KIM et al., 2021). A principal motivação visa atender a demanda cada vez maior dos usuários de *software* por recursos novos e inovadores lançados com frequência, além de desempenho e disponibilidade ininterruptos.

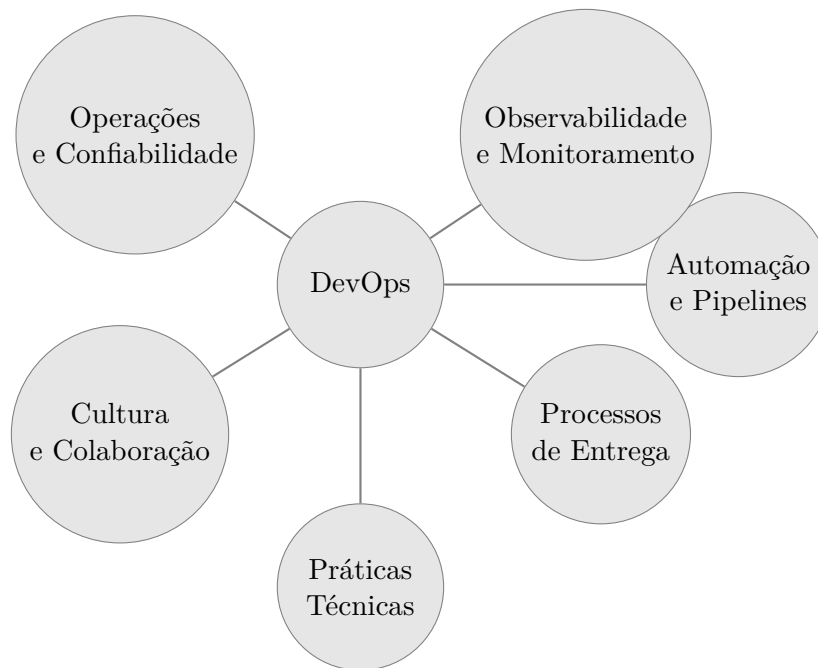


Figura 2.2: Mapa mental do DevOps: cultura, práticas, processos, automação, observabilidade e operações.

⁶Paradigma de gestão que considera qualquer sistema gerenciável como limitado em alcançar suas metas por um número reduzido de restrições.

⁷Sistema de produção desenvolvido pela Toyota entre 1947 e 1975, que aumenta a produtividade e a eficiência, evitando desperdícios sem criar estoques, como tempo de espera, superprodução e gargalos de transporte.

⁸Estudo de tecnologias tolerantes a falhas em sistemas.

⁹Técnica de aprendizagem adaptativa e generativa, que incentiva colaboradores a pensar novas formas e trabalhar em conjunto para encontrar a melhor resposta a qualquer problema.

¹⁰Conjunto de políticas, normas, valores, atitudes e pressupostos relacionados à segurança.

¹¹Diz respeito à integridade física, mental, emocional e moral dos colaboradores.

2.1.1 *Pipeline* em *DevOps*

Para garantir a padronização e a qualidade no desenvolvimento, além de tempestividade e foco na implementação, é imprescindível a existência de uma esteira, ou uma *pipeline*, termo comum utilizado na área.

Um conjunto de processos e ferramentas automatizadas representa a *pipeline*, onde se provê aos profissionais desenvolvedores e operadores uma maneira coesa de trabalho, desde a criação até a implementação do código em ambiente de produção. Embora possam existir diferenças nas *pipelines* de acordo com a organização, em geral inclui a automação da compilação/integração contínua, testes de automação, validação e geração de relatórios. A continuidade é uma característica diferenciada na *pipeline*. Inclui-se integração contínua, entrega/implementação contínua (CI/CD), *feedback* contínuo e operações contínuas. Em vez de testes pontuais ou implementações agendadas, cada função ocorre de forma contínua (CARTER et al., 2022).

O design e a implementação em uma organização depende do conjunto de ferramentas utilizadas, do nível de experiência dos engenheiros de *DevOps*, do orçamento e de outros fatores, como qualidade ou segurança. Para construção de uma *pipeline* é recomendado a atuação de profissionais de TI com amplo conhecimento das áreas de desenvolvimento, operações e segurança, incluindo codificação, gerenciamento de infraestrutura, administração de sistemas e cadeias de ferramentas. Convém destacar que este processo é contínuo, ou seja, a *pipeline* evolui conforme seu uso, maturidade, cultura e novas exigências.

São componentes de uma *pipeline*:

1. **Steps:** são fases ou estágios de uma *pipeline*. Em cada *step* acontece algum tipo de automatização, como compilação, armazenamento, teste ou qualquer tipo de ação que seja necessária. Tem como características principais: a dinamicidade, é possível versionar ou adicionar um novo *step* em qualquer ponto da *pipeline*; e o isolamento, executa-se de maneira isolada e sequencial (CARTER et al., 2022).
2. **Integração e entrega contínuas (CI/CD)**¹²: é a prática de fazer *commits*¹³ frequentes em um repositório de código-fonte. CI/CD integra continuamente as alterações de código em um repositório, para que conflitos entre diferentes versões providas pelos desenvolvedores sejam identificadas com rapidez e relativamente fáceis de corrigir (CARTER et al., 2022). Trata-se de uma prática necessária para aumentar a eficiência de implementação.

¹²Integração Contínua (*Continuous integration*) e Entrega Contínua (*Continuous Delivery*) são abordagens ágeis de desenvolvimento de *software* que visam automatizar o processo de construção, teste e entrega de *software*

¹³Registros históricos de entrega de um conjunto de mudanças provisórias ou permanentes para um repositório que armazena código-fonte de uma solução.

O desenvolvimento baseado em pequenas entregas é um requisito para a integração contínua. A ideia é aumentar o controle da evolução do código-fonte entregue. Se não houver *commits* frequentes em uma ramificação comum de um repositório de código-fonte compartilhado, não haverá integração contínua (KIM et al., 2021). Se os processos de compilação e testes forem automatizados, mas os desenvolvedores estiverem trabalhando em ramificações de recursos isoladas, e de longa duração, sem integração a uma ramificação compartilhada, não existirá a implementação contínua.

A entrega contínua garante que a ramificação principal ou secundária do código-fonte de um aplicativo esteja pronta. Em outras palavras, se existir a necessidade de publicar tempestivamente uma versão mais recente de uma solução, essa versão pode ser implementada de maneira breve, sem risco de falhas (FORSGREN; HUMBLE, 2016a). Isto significa ter um ambiente de pré-produção ou homologação o mais parecido possível com o ambiente de produção, e garantir que os testes automatizados sejam executados previamente, de modo que todas as variáveis que possam causar falhas sejam identificadas antes que o código seja mesclado na ramificação principal ou secundária (KIM et al., 2021).

A implementação contínua envolve níveis de testes e operações contínuos tão robustos que novas versões de uma solução podem ser validadas e implementadas em um ambiente de produção sem exigir nenhuma intervenção humana.

3. **Feedback contínuo:** corresponde ao retorno rápido sobre a qualidade da codificação, obtido manualmente por meio da revisão de código realizada por outros desenvolvedores ou de forma automatizada, via testes unitários, de integração, de sistema e monitoramento. Essa prática contribui para a melhoria da solução (KIM et al., 2021). No modelo de desenvolvimento em cascata, a ausência de *feedback* em tempo adequado fazia com que novos recursos, ao levar meses ou anos da ideia à implementação, frequentemente resultassem em divergências em relação às expectativas iniciais.

O teste contínuo constitui parte central da *pipeline*, assegurando qualidade e consistência nas entregas. Em processos de *DevOps*, as mudanças passam continuamente do desenvolvimento para os testes e a implementação, resultando em lançamentos mais rápidos e produtos com maior confiabilidade. Isso envolve a execução de testes automatizados, incluindo testes unitários em cada alteração de compilação, testes operacionais, funcionais e de ponta a ponta (KIM et al., 2021).

A abordagem *DevOps* também incorpora o monitoramento contínuo em ambientes de pré-produção, testes e desenvolvimento. O monitoramento desses ambientes, voltado à identificação de comportamentos inesperados, é necessário para avaliar de

forma constante a integridade e o desempenho das soluções em produção (FORSGREN; HUMBLE; KIM, 2018). Ferramentas e serviços permitem garantir o *feedback* contínuo em diferentes níveis, desde a infraestrutura local ou em nuvem — como recursos de servidor e rede — até o desempenho da solução e das interfaces de API.

4. **Operações contínuas:** o termo é relativamente novo e menos comum, e as definições variam. Uma forma de interpretá-lo é como o “tempo de operação contínuo”. Por exemplo, o caso de uma estratégia de implementação *blue/green*¹⁴ em que se tem dois ambientes de produção separados, um “*blue*” (acessível publicamente ou oficial) e outro “*green*” (restrito ou piloto). Nessa situação, um novo código seria implementado no ambiente “*green*”. Quando houver a confirmação do *status* de funcional, um balanceador de carga muda o tráfego para a versão “*blue*”. O resultado é a ausência de tempo de inatividade para os usuários finais. Isto é, nenhuma indisponibilidade da solução no momento de publicação da nova versão (KIM et al., 2021).

Outra forma de pensar em operações contínuas é como um alerta contínuo. Essa é a noção de que o time de engenharia está de plantão e é notificado em casos de anomalias de desempenho na solução ou infraestrutura. Na maioria das situações o alerta contínuo vem acompanhado do monitoramento contínuo.

¹⁴Modelo de lançamento de solução que transfere gradualmente o tráfego do usuário de uma versão anterior para um novo lançamento quase idêntico, ambos executados em produção.

2.2 Frameworks de Avaliação *DevOps*

A avaliação do desempenho em práticas de *DevOps* pode ser realizada por diferentes frameworks, cada qual com foco e natureza específicos. Entre os mais utilizados e discutidos na literatura recente destacam-se o DORA, o SPACE e o CALMS, que oferecem perspectivas complementares sobre métricas de desempenho, produtividade e cultura organizacional.

O **DORA** consolidou-se como referência internacional para a mensuração objetiva de desempenho, oferecendo métricas quantitativas que permitem comparações entre equipes e organizações (FORSREN; HUMBLE; KIM, 2018).

O **SPACE** amplia a análise ao considerar dimensões relacionadas à produtividade dos desenvolvedores, como satisfação, comunicação e eficiência. Por basear-se em percepções qualitativas, apresenta maior subjetividade e menor padronização entre diferentes contextos (FORSREN; STOREY; ZHOU, 2021).

O **CALMS** enfatiza princípios culturais e organizacionais, funcionando como guia conceitual para adoção de práticas de *DevOps*. Embora relevante para orientar mudanças culturais, não fornece métricas quantitativas de desempenho (HUMBLE; MOLESKY, 2020).

A Tabela 2.1 sintetiza as principais características desses frameworks, evidenciando dimensões, foco e limitações. A análise fundamenta a escolha do DORA como núcleo avaliativo deste trabalho, por oferecer indicadores objetivos e comparáveis, especialmente adequados a ambientes regulados e de grande escala.

Framework	Dimensões principais	Foco	Limitações
DORA	Velocidade (frequência de implantações, tempo médio para mudanças); Estabilidade (taxa de falhas, tempo de recuperação)	Mensuração objetiva de desempenho	Restrito a quatro métricas; não captura aspectos culturais ou organizacionais mais amplos
SPACE	Satisfação, Performance, Atividade, Comunicação, Eficiência	Produtividade de desenvolvedores	Mais subjetivo; depende de percepções e dados qualitativos, difícil padronização
CALMS	Culture, Automation, Lean, Measurement, Sharing	Princípios culturais e organizacionais	Não fornece métricas quantitativas; serve como guia conceitual

Tabela 2.1: Comparação entre frameworks de avaliação em *DevOps*

A comparação demonstra que cada framework possui natureza e objetivos distintos. O DORA destaca-se por oferecer métricas quantitativas e comparáveis, permitindo mensuração objetiva do desempenho. O SPACE contribui com fatores humanos e de colaboração, enquanto o CALMS orienta aspectos culturais e organizacionais. Nesse contexto, a escolha do DORA como núcleo avaliativo desta dissertação justifica-se pela capacidade de fornecer indicadores objetivos aplicáveis ao setor financeiro, considerando que os demais podem atuar de forma complementar.

2.3 Métricas DORA

As métricas surgiram a partir do trabalho de pesquisa *DevOps Research and Assessments* (DORA), uma organização fundada em 2014 pelos pesquisadores Dr. Nicole Forsgren, Jez Humble e Gene Kim. Em 2018, a empresa *Google* incorporou a organização e suas métricas à sua plataforma de nuvem. A estrutura de métricas foi criada para apoiar times *DevOps* na análise da eficiência com que desenvolvem, entregam e mantêm o *software* (FORSGREN; HUMBLE; KIM, 2018). O programa DORA se consolidou como um dos principais esforços de pesquisa sobre entrega de *software* e operações, produzindo evidências que permitem aos times direcionar práticas e observar resultados.

Os relatórios produzidos anualmente pela DORA e *Google Cloud* desde 2018 apresentam percepções do mercado sobre práticas de *DevOps* e métricas que buscam representar o estágio de adoção dessas práticas. A estrutura classifica times em grupos de desempenho e oferece uma referência para que organizações avaliem seus processos ao longo do tempo. Esses relatórios resultam de um conjunto amplo de pesquisas com profissionais do setor, consolidando um corpo de evidências que fundamenta o uso das métricas DORA (FORSGREN; HUMBLE; KIM, 2018).

Embora times possam fornecer percepções sobre seu próprio desempenho, as métricas DORA fornecem um mecanismo sistemático para analisar processos relacionados à entrega de *software*. São coletadas informações sobre cinco dimensões: frequência de implantações, tempo gasto para mudanças, tempo de recuperação, taxa de falhas e desempenho operacional. O objetivo é mensurar o estado atual do processo de entrega e fornecer uma base de comparação entre diferentes contextos (HUMBLE; KIM, 2018). A pesquisa oficial *DORA DevOps Quick Check* (RESEARCH; (DORA), 2023) oferece uma avaliação de referência baseada em múltipla escolha, permitindo análise comparativa entre organizações (FORSGREN; HUMBLE; KIM, 2018).

Os resultados da pesquisa DORA apresentam percepções gerais sobre práticas organizacionais. Por isso, algumas empresas recorrem a avaliações específicas para analisar cultura, práticas, tecnologia e processos. Essas avaliações buscam identificar pontos de

atenção nos fluxos de trabalho. A partir desse acompanhamento, os times passam a conduzir ajustes regulares em seus processos de *DevOps*, incorporando atividades de revisão, análise e experimentação. A evolução gradual dessas práticas demanda revisões contínuas de rotinas, responsabilidades e mecanismos de aprendizado (SONI, 2020).

O relatório “O estado do *DevOps*” (FORSGREN; HUMBLE; KIM, 2018) desde suas primeiras versões evidencia quatro métricas principais, conforme tabela 2.2. As métricas trazem duas visões: velocidade e estabilidade. A frequência de implantações e o tempo gasto para mudanças medem a velocidade. E as métricas de tempo de recuperação e a taxa de falhas medem a estabilidade. Usadas em conjunto, essas quatro métricas fornecem uma linha de base do desempenho do time e indícios sobre onde ela pode ser melhorada.

Métrica	Baixo	Médio	Alto
Frequência de implantações Qual frequência sua organização publica código em produção, ou lança para usuários finais?	Entre 1x ao mês ou a cada 6 meses	Entre 1x por semana ou mês	Sob demanda (múltiplas entregas por dia)
Tempo gasto para mudanças Quanto tempo custa desde a codificação até a entrega da solução em produção?	Entre um a seis meses	Entre uma semana a um mês	Entre um dia a uma semana
Tempo de recuperação Quanto tempo demora para se restaurar um serviço em produção?	Entre uma semana a um mês	Entre um dia a uma semana	Menos de um dia
Taxa de falhas Qual o percentual de entregas em produção que resultam em indisponibilidade?	46%-60%	16%-30%	0%-15%

Tabela 2.2: Métricas de Desempenho de Entrega de *Software*.

1. **Frequência de implantações:** indica a frequência com que uma organização implanta com êxito o código para produção ou libera *software* para usuários finais (frequência de entregas). O objetivo de desenvolvimento contínuo do *DevOps* requer essencialmente que os times alcancem várias implantações diárias. Esta métrica fornece uma visão clara de sua posição em relação a esse objetivo. Um alto indicativo demonstra que há entregas de valor aos usuários finais e às partes interessadas. Além disso, mais implantações oferecem mais oportunidades para melhorar e implementar refinamentos no código-fonte (FORSGREN; HUMBLE; KIM, 2018).
2. **Tempo gasto para mudanças:** mede o tempo médio entre a confirmação e a liberação de um versionamento de código para produção (tempo de ciclo). A mensuração desta métrica é necessária, pois quanto menor o tempo, mais rapidamente

a equipe pode receber *feedback* e implementar melhorias nas soluções. É calculado medindo quanto tempo leva para concluir cada história de usuário¹⁵ do início ao fim, e calculando a média desses tempos. Analisar e direcionar os obstáculos do tempo gasto para mudanças permite que times melhorem sua velocidade de implantação na produção (FORSGREN; HUMBLE; KIM, 2018).

3. **Tempo de recuperação:** mede o tempo médio necessário para reparar e restaurar a funcionalidade de uma solução. É uma métrica que mensura a disponibilidade de sistemas, equipamentos, aplicativos, infraestrutura e eficiência de recuperação (tempo médio de restauração). Um tempo mais curto para restaurar o serviço permite que seus usuários usem serviços mais rapidamente após uma falha, permitindo que eles utilizem de forma recorrente. A indisponibilidade pode ser de qualquer natureza, desde um defeito no código em produção ou uma interrupção de infraestrutura não planejada (FORSGREN; HUMBLE; KIM, 2018).
4. **Taxa de falhas:** indica a porcentagem de implantações que causam uma falha na produção, o que exige correção imediata, como degradação ou interrupção do serviço (taxa de falha em mudanças). Uma baixa taxa de falha em mudanças é desejável porque quanto mais tempo um time passa lidando com falhas, menos tempo se tem para entregar novos recursos e valor ao cliente. Essa métrica geralmente é calculada contando quantas vezes uma implantação resulta em insucesso e dividindo esse valor pelo número total de implantações para obter uma média (FORSGREN; HUMBLE; KIM, 2018).

2.3.1 Métricas de Velocidade

As primeiras métricas a qual geralmente os times atuam para início do uso das métricas DORA é a frequência de implantações e tempo gasto para mudanças. A motivação destas métricas é avaliar quantas vezes uma solução é versionada em produção e a média do tempo gasto. O objetivo de entregar o código rapidamente em produção é publicá-lo o máximo de vezes no menor tempo possível. Para viabilizar é preciso aplicar a estratégia de diminuir a quantidade de mudanças no código de cada pacote. Em outras palavras, enviar para a produção o mínimo possível por vez (KIM et al., 2021).

Um equívoco comum sobre a frequência de implantações é que, ao publicar em produção com mais frequência, cria-se mais riscos. O pensamento é que, se uma certa porcentagem das alterações em produção falha ou causa incidentes, se isso ocorrer com maior frequência, então existirão mais incidências. Mas, de forma contraintuitiva, funciona exa-

¹⁵Descrição informal em linguagem simples do que um usuário quer fazer dentro de um produto de *software* para obter algo que ele considere valioso.

tamente de maneira oposta, ou seja, quanto mais se publica em produção com mudanças menores, melhor é compreendida cada uma dessas mudanças. Quando essas mudanças são compreendidas e são de pequeno alcance, o risco de falhas é menor. Ao reduzir o tamanho das entregas e enviar com a maior frequência possível, reduz-se o risco geral (HUMBLE; KIM, 2018).

2.3.2 Métricas de Estabilidade

A taxa de falha em alterações é a proporção entre o número de implantações e o número de falhas (FORSGREN; HUMBLE; KIM, 2018). É necessário definir o que constitui uma falha, pois essa métrica pode variar conforme o contexto de cada equipe. O conceito de falha pode mudar com o tempo conforme o time for evoluindo a sua percepção. Um equívoco comum é simplesmente observar o número total de falhas em vez sua taxa (HUMBLE; KIM, 2018).

O foco de um time *DevOps* é enviar mudanças o mais rápido possível, mas se for acompanhado simplesmente o número total de falhas, a percepção e resposta natural é tentar reduzir o número de implantações para que se tenha menos incidências. O problema disso, como mencionado anteriormente, é que as mudanças se tornam complexas na qual o impacto de uma falha, quando ocorrer, será alto. Quando uma falha acontece é esperado que seja de simples complexidade e bem compreendida pelo o time para uma atuação tempestiva (HUMBLE; KIM, 2018).

O envolvimento técnico dos desenvolvedores em publicações recorrentes em ambiente de produção é indispensável para consolidar práticas de entrega contínua. Idealmente dominando a entrega prestes a ser implantada. Quando ocorre uma falha, os desenvolvedores devem compreender o impacto de sua alteração e da falha resultante, assegurando que possam extrair aprendizado desse processo. Criando um feedback crítico sem culpa, uma cultura em prol de consolidar as falhas para que a comunidade de desenvolvedores garanta que tipos específicos de incidentes não sejam recorrentes e evitados (KIM et al., 2021).

2.3.3 Desempenho Operacional

A partir de 2021, os pesquisadores do DORA introduziram uma nova medida voltada às práticas operacionais modernas: o **desempenho operacional**. Essa métrica é fundamentada na **confiabilidade**, entendida como a capacidade dos serviços em atender às expectativas dos usuários, abrangendo aspectos como disponibilidade, latência, desempenho e escalabilidade. Historicamente, a disponibilidade era utilizada como indicador principal; entretanto, por ser apenas um dos focos da engenharia de confiabilidade, optou-

se por expandir a avaliação para a confiabilidade, de modo a representar de forma mais ampla os atributos de qualidade.

Na pesquisa, os entrevistados foram convidados a classificar sua capacidade de atender ou exceder metas de confiabilidade. Os resultados revelaram que times com diferentes níveis de desempenho de entrega obtêm melhores resultados — como menor esgotamento e maior sustentabilidade — quando também priorizam o desempenho operacional (SMITH et al., 2021).

Para avaliação, o desempenho operacional é analisado em conjunto com as demais categorias das métricas DORA, conforme apresentado na Tabela 2.3:

- **Estabilidade:** composta pelo tempo para restaurar um serviço e pela taxa de falha de alteração;
- **Desempenho operacional:** representado pela confiabilidade;
- **Produtividade:** composta pelo tempo de execução de alterações de código e pela frequência de implantações.

Fase	Estabilidade		Desempenho operacional	Velocidade		% Cor-respondentes
	Tempo de recuperação	Taxa de falhas	Confiabilidade	Tempo de entrega	Frequência de implantações	
Iniciando	Entre 1 dia e 1 semana	31%-45%	Às vezes atende expectativas	Entre 1 semana e 1 mês	Entre 1 semana e 1 mês	28%
Performando	Menos de 1 hora	0%-15%	Geralmente atende expectativas	Menos de 1 dia	Sob demanda (múltiplas entregas por dia)	17%
Reduzindo	Menos de 1 dia	0%-15%	Geralmente atende expectativas	Entre 1 semana e 1 mês	Entre 1 semana a 1 mês	34%
Descontinuando	Entre 1 a 6 meses	46%-60%	Geralmente atende expectativas	Entre 1 semana e 1 mês	Entre 1 a 6 meses	21%

Tabela 2.3: Avaliação de Desempenho Operacional

A criação dessa métrica decorre da constatação de que o desempenho de entrega de *software* contribui substancialmente para o desempenho operacional geral da organização. Ao priorizar confiabilidade, evitam-se armadilhas comuns das métricas de *software*, como a competição entre funções ou otimizações locais que não refletem ganhos sistêmicos (SMITH et al., 2021).

2.4 Sobre a Instituição Financeira

Presente em três continentes, a instituição financeira objeto deste estudo possui 3.983 agências distribuídas em 10 países, contando com aproximadamente 86.000 funcionários. No Brasil, dispõe de 11.115 pontos de atendimento, entre agências e postos próprios. Desses, 95% contam com salas de autoatendimento, totalizando mais de 40 mil terminais que funcionam além do expediente bancário. A instituição também opera uma rede de quase 56.000 pontos de atendimento adicionais, composta por correspondentes bancários e bancos

Entre as diversas diretorias existentes, a área de tecnologia é responsável por sustentar os temas relacionados à infraestrutura e inovação, além de impulsionar a transformação digital, com foco nos times *DevOps*. Desde 2020, essa área passou a adotar princípios da metodologia ágil, por meio dos quais foram desenvolvidos e concluídos mais de 300 empreendimentos.

A estrutura tecnológica está organizada em dois segmentos principais: construção e infraestrutura, conforme descrito a seguir:

1. **Construção:** duas unidades de engenharia e desenvolvimento dedicadas à criação e atualização de sistemas em parceria com as áreas de negócios, visando disponibilizar produtos e serviços com agilidade, qualidade e valor agregado ao cliente.
2. **Carreira tecnológica:** papéis estruturados em três níveis de assessoramento, com possibilidade de evolução pela chamada carreira em Y, que permite ao profissional optar por trajetórias técnicas ou gerenciais.
3. **Infraestrutura:** unidade de operações e soluções responsável por todo o parque tecnológico, incluindo centros de operações localizados em Boston, Paris e Amsterdam.

Os objetivos da área de tecnologia incluem proporcionar experiências digitais sustentáveis e inovadoras, além de acelerar a transformação dos processos organizacionais para ampliar a geração de valor. Por ser uma das maiores diretorias da instituição, sua atuação abrange ampla gama de funcionalidades e responsabilidades, entre as quais destacam-se:

1. Prospectar soluções tecnológicas para negócios digitais, incluindo processos e métodos de arquitetura de sistemas.
2. Definir processos corporativos, critérios de identificação de problemas, normas de funcionamento dos sistemas em ambientes de desenvolvimento, homologação e produção; além de arquiteturas de soluções digitais e aplicativos, integração de aplicações, iniciativas *Business-to-Business* (*Business-to-Business* (B2B)), modelos de interação digital com clientes e métodos de desenvolvimento.

3. Realizar testes de sistemas em projetos, incluindo testes de performance, estresse e carga; implantar, atualizar, manter e descontinuar soluções; liberar serviços e automatizar infraestrutura; além de garantir disponibilidade de serviços.
4. Desenvolver soluções de infraestrutura e negócios digitais; implementar e avaliar resultados; automatizar processos de segurança (identidade, autenticação, autorização, controle de acesso, criptografia); apoiar a transformação de serviços de aplicativos e disponibilização via APIs; gerir processos relacionados ao Catálogo de Serviços e atendimento de incidentes.
5. Implementar soluções para demandas internas, como conectividade lógica de equipamentos, liberação de serviços, modelos de qualidade em processos e produtos, repositórios de dados e conciliação de licenças de *software*.
6. Manter o planejamento estratégico, modelos de arquitetura, métodos de desenvolvimento, soluções de comunicação, parâmetros de orçamentação, gestão de portfólio e iniciativas; além de atividades de gestão de pessoas, apoio logístico e administrativo, manutenção de sistemas e aplicativos e treinamentos presenciais e on-line.
7. Gerir riscos tecnológicos, projetos estruturantes, catálogo de serviços, recursos humanos, orçamento e despesas, investimentos, eficiência operacional, metodologias de medição de esforço, performance de recursos e serviços, governança e inteligência competitiva.
8. Administrar ferramentas de segurança, comissões técnicas, adoção e evolução de plataformas corporativas, processos de mudança e liberação, gestão de problemas, testes não funcionais e homologação, configuração e disponibilidade de serviços, incidentes e capacidade, além de desenvolvimento de soluções em fábrica de *software*.
9. Elaborar estimativas de esforço para desenvolvimento e infraestrutura, requisitos de negócio e sistemas, cenários de testes, acordos de nível de serviço, planos de construção e aquisição, topologias de infraestrutura, padrões de automação e integração, planos de capacidade e desempenho e diretrizes de arquitetura.
10. Formular o Plano Estratégico de Tecnologia, relatórios de mudanças e liberações, acompanhamento da metodologia de atendimento digital, planejamento e integração de soluções, monitoramento e controle de informações.
11. Definir modelos de referência de arquitetura, estudos e tendências de mercado, impactos de custos, metodologias de avaliação de desempenho e técnicas de inteligência cognitiva.

12. Monitorar componentes de infraestrutura e soluções, padrões e diretrizes de arquitetura, revisão e integração de processos, implementação de planos de comunicação, aderência a modelos de gestão de processos, identificação e mitigação de riscos e manutenção de controles de conformidade.
13. Avaliar portfólio de soluções, resultados de desenvolvimento, requisições de mudanças e liberações, desempenho de processamento, qualidade em processos e produtos, custos comparativos, viabilidade técnica e econômica, propostas de portfólio e capacidade de atendimento.

2.5 Trabalhos Correlatos

A literatura sobre *DevOps* e métricas DORA evoluiu de forma significativa na última década. Para dar clareza à jornada histórica, os trabalhos correlatos são apresentados em ordem cronológica, divididos em períodos que marcam momentos importantes da evolução. Essa organização permite compreender como os conceitos se consolidaram, como foram aplicados em grandes empresas e instituições financeiras, e como se relacionam com a lacuna científica abordada nesta dissertação.

Sobre *DevOps* em Grandes Empresas

A literatura sobre *DevOps* em grandes empresas foi organizada em períodos cronológicos para evidenciar a jornada de evolução da prática. Essa divisão permite compreender como os fundamentos culturais e técnicos foram estabelecidos, como os casos de sucesso consolidaram a adoção, como o paradigma se expandiu para lidar com sistemas legados e, finalmente, como os avanços recentes trouxeram aplicações em setores críticos como o financeiro e estratégias de escala em corporações globais. Cada período marca um momento importante da maturidade do *DevOps* e ajuda a contextualizar os desafios enfrentados pela instituição foco desta

Período Inicial (2016–2018)

Forsgren e Humble (2016b) introduzem o modelo CALMS (cultura, automação, *lean*, métricas e compartilhamento), que se tornou referência para avaliar a prontidão organizacional para adoção do DevOps. O modelo enfatiza que a transformação não pode ser apenas técnica: é necessário alinhar cultura organizacional, práticas de automação, princípios enxutos de melhoria contínua, métricas de desempenho e mecanismos de compartilhamento de conhecimento. O CALMS passou a ser utilizado como um guia de maturidade, permitindo que gestores avaliem se suas organizações estão preparadas para

a adoção do DevOps e quais dimensões precisam ser fortalecidas. Esse trabalho é considerado um marco inicial porque sistematiza os elementos fundamentais que sustentam a prática.

Hering (2018) apresenta o conceito de conhecimento em forma de T (*T-shaped skills*), destacando a necessidade de profissionais com competências amplas e especialização profunda em uma área. A parte superior do “T” representa o conhecimento generalista, enquanto a haste vertical representa a especialização. Essa abordagem busca superar o problema da alta especialização isolada, incentivando que os profissionais entendam como seu trabalho impacta outras áreas da pilha técnica. O estudo reforça que o sucesso do DevOps depende de times capazes de colaborar transversalmente, com compreensão ampla de processos e tecnologias, mas também com domínio profundo em áreas críticas. Essa visão contribui para a formação de times multifuncionais e resilientes.

Em estudo posterior, Forsgren (2018) discutem os desafios culturais e técnicos da implementação de pipelines de entrega contínua. O trabalho mostra que, embora a automação seja essencial, ela não é suficiente sem mudanças culturais que promovam colaboração, confiança e responsabilidade compartilhada entre desenvolvimento e operações. Os autores destacam dificuldades comuns enfrentadas por organizações, como resistência cultural, falta de liderança transformacional e ausência de métricas claras para avaliar progresso. Além disso, apontam que a implementação de pipelines requer não apenas ferramentas, mas também práticas de governança e alinhamento estratégico. Esse estudo é relevante porque evidencia que a adoção do DevOps não é apenas uma questão de tecnologia, mas de mudança organizacional profunda.

No período inicial (2016–2018), consolidaram-se os fundamentos culturais e técnicos que sustentam o DevOps. O modelo CALMS forneceu uma estrutura conceitual, o conhecimento em forma de T trouxe uma abordagem para capacitação dos profissionais, e os estudos sobre pipelines de entrega contínua mostraram os desafios práticos da implementação. Esses trabalhos criaram a base teórica e prática que orientou os casos de sucesso posteriores em grandes empresas.

Expansão e Casos de Sucesso (2019–2021)

Wiedemann et al. (2019) relatam histórias de sucesso em empresas como Kaiser Permanente, Capital One, Target, Starbucks e ING, que conseguiram reduzir drasticamente o tempo de entrega de software para aplicativos críticos. O estudo mostra que a combinação entre desenvolvimento e operações permitiu que novos recursos fossem disponibilizados em questão de segundos, aumentando a satisfação dos clientes e a rentabilidade das organizações. Além dos ganhos técnicos, os autores destacam melhorias culturais, como maior colaboração entre departamentos e melhor equilíbrio entre vida pessoal e profissional dos

integrantes dos times. O trabalho evidencia que a adoção do DevOps não apenas acelera a entrega de valor, mas também promove maturidade organizacional e engajamento dos profissionais.

Na instituição financeira analisada, a busca por maior frequência de implantações também foi relatada como prioridade, mas os sistemas legados e a burocracia regulatória ainda limitam a obtenção dos mesmos resultados descritos nos casos internacionais.

Lwakatare et al. (2019) analisam cinco grandes empresas e identificam que a cultura de confiança, capacitação dos times e automação são pilares fundamentais da adoção do DevOps. O estudo mostra que a remoção de etapas manuais e burocráticas nos processos de implantação aumentou a produtividade e reduziu falhas humanas. Os autores destacam ainda a importância das comunidades internas de conhecimento, chamadas de *Chapters*, que promovem aprendizado contínuo e disseminação de boas práticas. Essa abordagem reforça que o DevOps não é apenas uma mudança técnica, mas também cultural, baseada em cooperação, autonomia e compartilhamento de informações. O trabalho é relevante porque demonstra que a adoção bem-sucedida depende de práticas organizacionais que sustentem a automação e a entrega contínua.

Smith et al. (2021) reforça o papel da entrega contínua como fenômeno de transformação digital. O autor argumenta que o DevOps se tornou um elemento central para organizações modernas que utilizam software como meio de entregar valor aos clientes, incluindo setores como bancos, varejo e atacado. O estudo mostra que quase todos os aspectos do ciclo de vida de desenvolvimento podem ser automatizados, permitindo maior frequência de implantações e redução de falhas. Smith também destaca que o DevOps marca uma mudança de paradigma: de projetos pontuais para produtos em evolução contínua. Essa visão conecta o DevOps à transformação digital mais ampla, em que a agilidade e a capacidade de adaptação são essenciais para a competitividade.

Entre 2019 e 2021, grandes empresas relatam ganhos concretos com a adoção do DevOps, incluindo velocidade de entrega, satisfação do cliente e maturidade organizacional. Os estudos mostram que a cultura de confiança e automação são pilares da prática, e que a entrega contínua se consolidou como fenômeno de transformação digital. Esse período marca a transição do DevOps de iniciativas isoladas para práticas escaladas em corporações globais.

Paradigma e Legados (2022)

Ebert e Hochstein (2022) caracterizam o DevOps como uma mudança de paradigma que impacta todo o setor de TI. O estudo mostra que, ao integrar práticas *lean* e ágeis, o DevOps promove automação de ponta a ponta no desenvolvimento e entrega de software, criando uma nova forma de pensar e organizar o trabalho. Essa abordagem rompe com o

modelo tradicional de silos entre desenvolvimento e operações, estabelecendo colaboração contínua desde os requisitos até a manutenção e evolução das soluções. Os autores destacam que essa integração resulta em ganhos expressivos: redução de até 50% no tempo de ciclo de entrega e diminuição de custos em até 20%, principalmente em ambientes centralizados de TI.

A figura 2.3 representa o que sucesso do modelo se deve a múltiplos fatores, incluindo arquitetura, cultura, ferramentas e habilidade (EBERT; HOCHSTEIN, 2022).

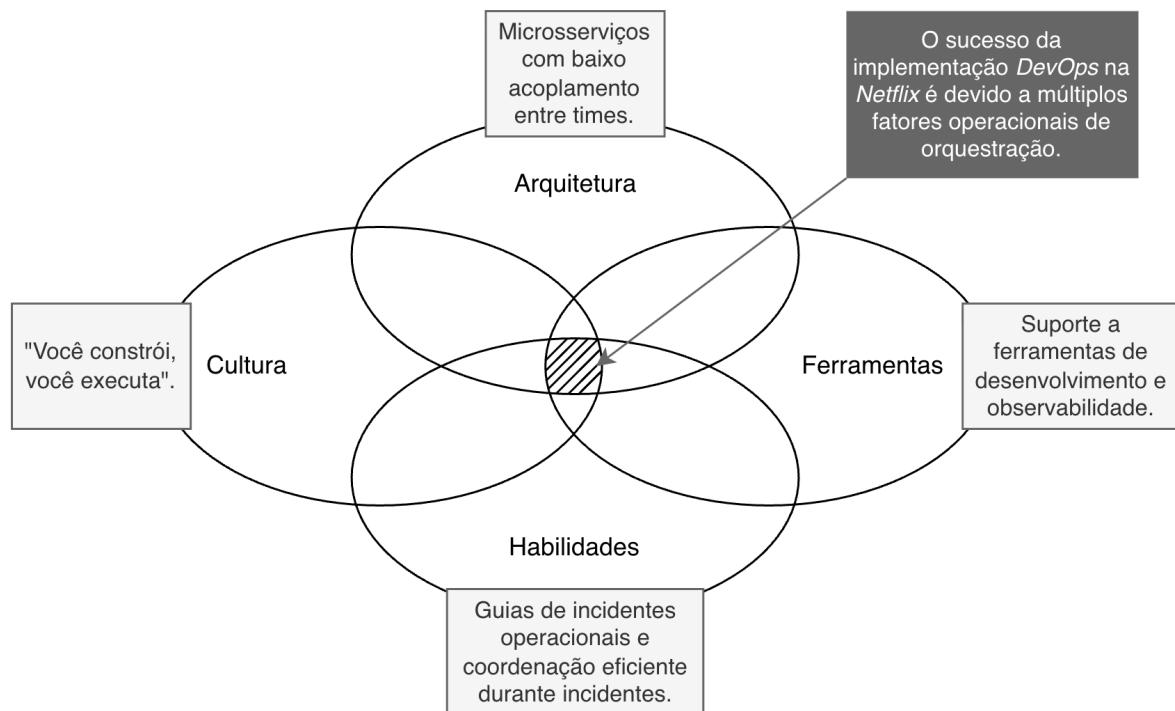


Figura 2.3: Modelo de Sucesso *DevOps* na *Netflix*, integrando cultura, arquitetura, ferramentas e habilidades.

Um ponto central do trabalho é a análise da complexidade envolvida na transformação de sistemas legados. Ebert e Hochstein (2022) argumentam que, embora os serviços nativos de nuvem se adaptem mais facilmente às práticas DevOps, grande parte das organizações ainda depende de arquiteturas antigas e fortemente acopladas. Essa dependência aumenta o custo e a dificuldade de implementar entrega contínua, exigindo soluções personalizadas e maior esforço de engenharia. O estudo mostra que a modernização de sistemas legados não pode ser tratada como uma simples migração técnica, mas como uma transformação organizacional que envolve processos, cultura e governança.

Esse ponto é diretamente aplicável ao contexto da instituição em estudo, cujo parque tecnológico é majoritariamente composto por sistemas legados, tornando a transformação DevOps mais complexa e custosa.

Os autores também apresentam uma visão geral das melhores práticas de DevOps, destacando que não existe uma “receita única” aplicável a todas as empresas. Cada organização precisa adaptar os princípios às suas necessidades específicas, considerando fatores como criticidade dos sistemas, requisitos de segurança e arquitetura tecnológica. Exemplos como o caso da Netflix são utilizados para ilustrar como diferentes modelos de entrega podem ser aplicados com sucesso, desde ambientes altamente centralizados até sistemas críticos que exigem runbooks e arquiteturas dedicadas para garantir segurança e confiabilidade.

Em 2022, o DevOps é reconhecido não apenas como uma prática técnica, mas como um paradigma organizacional capaz de transformar todo o setor de TI. O trabalho de Ebert et al. (EBERT; HOCHSTEIN, 2022) evidencia que os ganhos de desempenho e redução de custos são significativos, mas que o enfrentamento dos sistemas legados continua sendo um dos maiores desafios para a adoção plena. Essa discussão conecta-se diretamente ao contexto da instituição foco desta dissertação, cujo parque tecnológico é majoritariamente composto por sistemas legados, reforçando a relevância da pesquisa em propor soluções práticas para medir e evoluir a maturidade DevOps nesse cenário.

Avanços Recentes (2023–2025)

Bohanec e Morelli (2023) analisam práticas de melhoria contínua em grandes corporações, destacando que o *DevOps* não deve ser visto como uma iniciativa pontual, mas como um processo evolutivo e permanente. O estudo mostra que empresas que adotaram ciclos iterativos de avaliação e ajuste conseguiram reduzir gargalos, aumentar a confiabilidade dos sistemas e melhorar a satisfação dos clientes. Os autores enfatizam que a melhoria contínua depende de métricas claras, feedback rápido e cultura organizacional que valorize aprendizado e experimentação. Esse trabalho reforça a ideia de que o *DevOps* é uma jornada de maturidade, e não apenas a implementação de ferramentas.

Amaro, Pereira e Silva (2023) apresentam uma revisão multivocal sobre capacidades e práticas em *DevOps*, destacando que a adoção bem-sucedida depende tanto de fatores técnicos quanto culturais. O estudo evidencia que métricas como as DORA são fundamentais para avaliar desempenho organizacional e orientar a evolução das práticas, especialmente em ambientes complexos.

Jayakody e Wijayanayake (2023) identificam fatores críticos de sucesso para a adoção do *DevOps* em desenvolvimento de sistemas de informação. Entre os principais fatores estão o suporte da liderança, a cultura colaborativa e o uso de métricas de desempenho para monitorar resultados. O trabalho reforça que a maturidade em *DevOps* exige alinhamento organizacional e clareza na definição de indicadores.

Kose (2024) exploram a adoção do *DevOps* em desenvolvimento de software móvel, mostrando que práticas de automação e integração contínua são aplicáveis também em ambientes de alta volatilidade tecnológica. O estudo demonstra que, mesmo em setores emergentes, as métricas DORA podem ser utilizadas para avaliar desempenho e orientar melhorias.

Sanjurjo et al. (2024) avaliam o desempenho do *BizDevOps* utilizando padrões internacionais, com estudos de caso que evidenciam lições aprendidas em grandes organizações. Os autores destacam que a integração entre negócios, desenvolvimento e operações requer métricas claras e processos de governança, sendo as métricas DORA instrumentos relevantes para medir progresso e justificar investimentos.

No caso da instituição financeira em estudo, os achados desses trabalhos se mostram convergentes em diferentes dimensões. A necessidade de ciclos iterativos de melhoria e de métricas padronizadas, apontada por Amaro, Pereira e Silva (2023) e Amaro, Pereira e Silva (2023), reflete diretamente os desafios enfrentados para superar gargalos de integração e justificar investimentos em transformação digital. Da mesma forma, os fatores críticos de sucesso descritos por Jayakody e Wijayanayake (2023) — liderança engajada e cultura colaborativa — foram identificados como barreiras internas que ainda precisam ser superadas.

Os estudos de Kose (2024) e Sanjurjo et al. (2024) reforçam a pertinência da adoção de práticas de automação e integração contínua nos canais digitais da instituição, além da necessidade de alinhar processos de TI às exigências regulatórias e de compliance. Esses aspectos confirmam que os desafios enfrentados pela organização brasileira não são isolados, mas compartilham características comuns às grandes corporações globais.

Entre 2023 e 2025, o *DevOps* se consolidou em setores críticos, como o financeiro, e enfrentou o desafio de escalar em grandes corporações. Os estudos mostram que a melhoria contínua, a cultura organizacional, a superação de legados e compliance, e a adoção de métricas claras são fatores decisivos para o sucesso. Esse período marca a transição do *DevOps* para um padrão global de avaliação e prática, reforçando a relevância da implementação das métricas DORA no contexto brasileiro de instituições financeiras.

Sobre Métricas DORA

Os trabalhos sobre métricas DORA foram igualmente divididos em períodos cronológicos para mostrar a evolução da medição de desempenho em práticas *DevOps*. Essa separação evidencia como surgiram os primeiros fundamentos, como as métricas foram consolidadas e validadas empiricamente, como se expandiram para diferentes contextos organizacionais e, por fim, como se tornaram padrão global com guias práticos de implementação. Essa jornada histórica reforça a importância das métricas DORA como referência internacional

e destaca a lacuna científica desta dissertação: a implementação prática dessas métricas em instituições financeiras brasileiras.

Fundamentos (2013–2017)

Roche (2013) apresenta os primeiros relatos de adoção de métricas DevOps em ambientes corporativos. O trabalho discute como organizações começaram a medir aspectos básicos da entrega de software, como frequência de implantações e tempo de ciclo, para avaliar a maturidade de suas práticas. Embora ainda incipientes, essas métricas abriram caminho para a sistematização da avaliação de desempenho em DevOps, mostrando que medir era essencial para justificar investimentos e orientar melhorias.

Na instituição em estudo, percebe-se que a medição ainda é incipiente, com indicadores mais voltados a disponibilidade e incidentes, sem foco em métricas de fluxo como frequência de implantações ou tempo de ciclo.

McCarthy et al. (2015) discute arquiteturas componíveis como suporte à adoção do DevOps. O autor argumenta que sistemas construídos de forma modular permitem maior flexibilidade e velocidade na entrega de novos recursos. Essa visão antecipa a importância dos microsserviços e da arquitetura desacoplada, que se tornariam pilares do DevOps nos anos seguintes. O estudo mostra que a arquitetura não é apenas técnica, mas estratégica, pois influencia diretamente a capacidade de escalar práticas de entrega contínua.

Fowler (2016) aborda práticas de produção contínua, destacando que o objetivo do DevOps é reduzir o tempo entre a concepção de uma ideia e sua disponibilização em produção. O autor enfatiza que a produção contínua exige automação em todas as etapas do ciclo de vida, desde testes até implantação, e que a ausência de automação gera gargalos e riscos. Esse trabalho é relevante porque conecta diretamente a prática DevOps ao valor entregue ao cliente, reforçando que a velocidade e a confiabilidade são métricas centrais.

Esse conceito dialoga com a necessidade da instituição de reduzir o tempo entre concepção e entrega, especialmente em sistemas críticos de pagamento e crédito.

Erich, Amrit e Daneva (2017) exploram a qualidade em ambientes DevOps, mostrando que a integração entre desenvolvimento e operações melhora não apenas a velocidade, mas também a confiabilidade das entregas. O estudo qualitativo evidencia que práticas como testes automatizados e monitoramento contínuo reduzem falhas e aumentam a satisfação dos clientes. John et al. (2017) discute a aplicação do DevOps em serviços, destacando que a abordagem não se limita a produtos de software, mas pode ser aplicada a qualquer serviço digital que dependa de ciclos rápidos de atualização. Artac et al. (2017) analisa práticas DevOps em diferentes contextos, mostrando que a adoção requer adaptação às necessidades específicas de cada organização, mas que princípios como automação, colaboração e métricas são universais.

Entre 2013 e 2017 surgem os primeiros fundamentos para medir desempenho de entrega em DevOps. Os trabalhos desse período mostram que a adoção depende de métricas, arquitetura modular, automação e práticas de qualidade. Essa fase inicial consolidou os conceitos que seriam expandidos nos anos seguintes, preparando o terreno para a consolidação das métricas DORA como padrão global.

Consolidação (2018–2019)

Forsgren e Kersten (2018) e Forsgren, Humble e Kim (2018) consolidam o uso das métricas DORA como indicadores de desempenho organizacional. Os estudos mostram que quatro métricas — frequência de implantações, tempo de entrega de mudanças (*lead time*), taxa de falhas em mudanças e tempo médio de recuperação (MTTR) — são preditores robustos do desempenho de times de software. A pesquisa evidencia que organizações que alcançam altos níveis nessas métricas apresentam maior lucratividade, participação de mercado e satisfação dos clientes. Essa consolidação foi decisiva porque transformou as métricas DORA em padrão de referência global, permitindo que empresas comparassem sua maturidade DevOps de forma objetiva.

Na instituição, a ausência de métricas padronizadas dificulta a comparação entre times. A adoção das quatro métricas DORA poderia permitir avaliar maturidade de forma objetiva e identificar gargalos em sistemas legados.

Trihinas et al. (2018) aplica métricas em estudos empíricos para avaliar a eficiência de pipelines de entrega contínua. O autor mostra que a coleta sistemática de dados sobre tempo de execução, falhas e recuperação permite identificar gargalos e otimizar processos. Torble (2019) analisa a taxa de falhas em mudanças (*change failure rate*), destacando que essa métrica é essencial para avaliar a confiabilidade das entregas e orientar investimentos em testes automatizados e práticas de qualidade. Šćekić et al. (2018) discute a aplicação das métricas em diferentes contextos organizacionais, mostrando que, embora os valores absolutos variem, os indicadores são úteis para comparar times e identificar áreas de melhoria. Esses trabalhos reforçam a importância da validação empírica das métricas, demonstrando sua aplicabilidade em ambientes reais.

Katal, Bajoria e Dahiya (2019) contribui com benchmarks de adoção do DevOps, mostrando como diferentes setores industriais apresentam níveis variados de maturidade. O estudo evidencia que empresas de tecnologia tendem a alcançar métricas mais avançadas, enquanto setores tradicionais enfrentam maiores desafios. Esse tipo de benchmark é relevante porque a instituição busca se posicionar frente a concorrentes do setor financeiro, mas carece de dados comparativos internos e externos.

Haindl e Plösch (2019) propõe frameworks para medir desempenho DevOps, integrando métricas técnicas e organizacionais. O trabalho destaca que a medição não deve

se limitar a indicadores de entrega, mas também incluir aspectos culturais e de colaboração.

Bezemer et al. (2019) reforça a análise de desempenho, mostrando que a coleta contínua de métricas permite identificar padrões de falhas e prever riscos operacionais. O estudo conecta métricas DORA a práticas de observabilidade, ampliando sua relevância para a confiabilidade dos sistemas.

Entre 2018 e 2019, as métricas DORA deixaram de ser apenas conceitos teóricos e passaram a ser aplicadas em larga escala, validadas empiricamente e integradas a benchmarks de mercado. Esse período consolidou as métricas como padrão global de avaliação de desempenho DevOps, criando a base para sua adoção em diferentes setores e contextos organizacionais.

Expansão e Diversificação (2020–2022)

Humble, Molesky e O'Reilly (2020) conecta as métricas DORA às práticas *lean*, mostrando que a medição de desempenho em DevOps deve ser integrada a princípios de melhoria contínua e eliminação de desperdícios. O autor argumenta que as métricas não são apenas indicadores técnicos, mas instrumentos de gestão que permitem identificar gargalos, reduzir retrabalho e aumentar o fluxo de valor entregue ao cliente. Essa conexão entre DORA e *lean* reforça a ideia de que a transformação DevOps é parte de uma jornada maior de eficiência organizacional.

Na instituição, práticas de melhoria contínua são aplicadas em áreas de negócio, porém ainda não estão integradas às métricas de TI, o que restringe a visão sobre o fluxo de valor.

Complementando essa perspectiva, Maroukian e Gulliver (2020a) discute o papel da liderança na adoção do DevOps, destacando que líderes precisam criar ambientes de confiança e incentivar a autonomia dos times. O estudo mostra que a liderança transformacional é essencial para sustentar práticas técnicas e culturais, garantindo que as métricas sejam usadas para aprendizado e não para punição. Rosenthal e Jones (2020) aprofunda essa discussão ao introduzir o conceito de *chaos engineering* como prática complementar ao DevOps, defendendo que a simulação de falhas em ambientes controlados aumenta a resiliência dos sistemas e fornece dados valiosos para as métricas de confiabilidade. Em outra contribuição, Maroukian e Gulliver (2020b) reforça a importância de conectar métricas técnicas a indicadores de negócio, evidenciando que o impacto do DevOps só se concretiza quando alinhado a objetivos estratégicos da organização.

Essas abordagens ampliam a visão do DevOps, incluindo dimensões de liderança, resiliência e alinhamento estratégico.

López-Fernández et al. (2021) complementa esse panorama ao analisar a adoção do DevOps em diferentes setores, mostrando que práticas como automação de testes e obser-

vabilidade são fundamentais para reduzir falhas e aumentar a confiança nas entregas. Em linha com essa visão, Singh et al. (2021) discute a relação entre métricas DORA e cultura organizacional, destacando que times com maior autonomia e colaboração apresentam melhores resultados. Os relatos da instituição confirmam essa correlação, evidenciando que times mais autônomos têm desempenho superior nas métricas de fluxo e confiabilidade.

No campo da avaliação de maturidade, Sallin et al. (2021) propõe métodos que integram métricas técnicas e culturais em modelos de diagnóstico. O estudo mostra que medir apenas indicadores técnicos é insuficiente para capturar o verdadeiro impacto da transformação DevOps. Esse ponto é diretamente aplicável à instituição em estudo, que ainda carece de um modelo formal de avaliação de maturidade.

Por fim, Kim et al. (2021) reforça a importância da aprendizagem contínua e da experimentação, mostrando que organizações que incentivam inovação e tolerância ao erro conseguem melhorar suas métricas de entrega e confiabilidade. Esses trabalhos evidenciam que a expansão do DevOps envolve não apenas práticas técnicas, mas também dimensões culturais e organizacionais, exigindo abordagens integradas e adaptativas.

Entre 2020 e 2022, as métricas DORA foram ampliadas e diversificadas, conectando-se a práticas *lean*, liderança transformacional, *chaos engineering* e modelos de maturidade. Esse período marca a transição do DevOps de uma prática técnica para uma transformação organizacional completa, envolvendo cultura, estratégia e resiliência. A expansão reforça a necessidade de abordagens que considerem tanto indicadores técnicos quanto culturais, preparando o terreno para a consolidação das métricas como padrão global nos anos seguintes.

Avanços Recentes (2023–2025)

Ruiz et al. (2023) atualiza benchmarks de adoção das métricas DORA, mostrando como diferentes setores industriais evoluíram em termos de frequência de implantações, tempo de ciclo e confiabilidade. O estudo evidencia que organizações que adotaram práticas de automação e observabilidade avançada obtiveram ganhos significativos em velocidade e qualidade. Essa análise é relevante para a instituição em estudo, que busca comparar seu desempenho DevOps com benchmarks internacionais, mas ainda carece de dados internos estruturados para tal comparação.

Wilkes, Milani e Storey (2023) discute a evolução das métricas DORA, destacando que, além dos quatro indicadores tradicionais (frequência de implantações, tempo de entrega, taxa de falhas e tempo médio de recuperação), novas dimensões começam a ser consideradas, como segurança e experiência do desenvolvedor. Essa ampliação mostra que o DevOps está se tornando mais abrangente, conectando métricas técnicas a indicadores de valor de negócio. No contexto da instituição financeira analisada, essa evolução é

especialmente relevante, pois a segurança e a confiabilidade constituem requisitos críticos em sistemas bancários.

Na instituição, a discussão sobre a evolução das métricas ocorre em razão do interesse em migrar de indicadores tradicionais de TI para métricas voltadas ao fluxo e à confiabilidade.

O relatório oficial da Google Cloud / DORA (2025) consolida as métricas como padrão global de avaliação de desempenho DevOps. O documento mostra que as quatro métricas originais continuam sendo preditores robustos de desempenho organizacional, mas reforça a necessidade de instrumentação prática para coleta e análise contínua. Para a instituição em estudo, esse relatório é fundamental, pois valida internacionalmente o uso das métricas DORA como referência e justifica sua adoção como ferramenta de avaliação da maturidade DevOps.

A Atlassian (2025) oferece um guia prático de implementação das métricas DORA, mostrando como integrá-las a pipelines de entrega contínua e ferramentas de observabilidade. O material destaca estratégias para reduzir gargalos, aumentar a confiabilidade e alinhar métricas técnicas a objetivos de negócio. Checkmarx (2025) complementa essa visão, apresentando práticas para operacionalizar as métricas em ambientes complexos, com foco em segurança e qualidade. Ambos os guias são diretamente aplicáveis ao contexto da instituição financeira brasileira, que busca implementar APIs para calcular métricas DORA e integrar esses indicadores a seus processos de governança e compliance.

Esses guias práticos são diretamente aplicáveis ao contexto da instituição, que busca operacionalizar métricas por meio de APIs e integrações com pipelines de entrega.

No caso da instituição financeira em estudo, os achados desses trabalhos se mostram convergentes em diferentes dimensões. A atualização de benchmarks apresentada por Ruiz et al. (2023) é particularmente relevante, pois permite comparar a maturidade local com padrões internacionais. A ampliação das métricas discutida por Wilkes, Milani e Storey (2023) reforça a necessidade de incluir dimensões de segurança e confiabilidade, aspectos críticos em sistemas bancários. O relatório oficial da Google Cloud / DORA (2025) valida a adoção das métricas como referência global, enquanto os guias práticos de Atlassian (2025) e Checkmarx (2025) oferecem caminhos concretos para operacionalizar indicadores em pipelines e processos de compliance.

Entre 2023 e 2025, as métricas DORA se consolidaram como padrão global e passaram a ser foco de guias práticos de aplicação. Os estudos mostram que, além de medir desempenho técnico, as métricas evoluíram para incluir dimensões de segurança e valor de negócio. Para a instituição em estudo, esse período é decisivo, pois fornece evidências e ferramentas que sustentam a implementação prática das métricas DORA em seu ambiente, preenchendo a lacuna científica identificada nesta dissertação.

Capítulo 3

Metodologia

Neste trabalho adota-se a pesquisa aplicada como abordagem metodológica, visando à produção de conhecimento científico aliado à aplicação prática das métricas DORA na instituição financeira objeto do estudo. Essa estratégia permite avaliar o nível de desempenho dos times *DevOps* de forma estruturada e contextualizada.

Para o alcance dos objetivos, serão utilizados métodos quantitativos e qualitativos, conforme a natureza das análises. Nos estudos quantitativos, pretende-se realizar análises de conteúdo fundamentadas em números e estatísticas, atribuindo significado aos dados coletados, como informações sobre comportamento e desempenho dos times *DevOps*. O foco recai sobre medições que possibilitem comparações e agregações estatísticas consistentes (PATTON, 2002).

Os estudos qualitativos, por sua vez, serão conduzidos em nível descritivo, buscando compreender o significado explícito dos dados. Nessa abordagem, utilizam-se extratos de informações como amostras, em vez de análises exaustivas de todo o conjunto (MERRIAM; TISDELL, 2015). O objetivo é enfatizar a interpretação de opiniões de autores de artigos revisados, o feedback obtido a partir do uso das funcionalidades de extração das métricas DORA, bem como os relatos de especialistas e de integrantes dos times *DevOps* em entrevistas.

O desenvolvimento da pesquisa considera o cumprimento das seguintes atividades:

Fase	Objetivo	Atividade	Método/Técnica	Resultado
Fase 1	OE1 – Fundamentar o assunto <i>DevOps</i> .	A1 – Revisar literatura.	<i>Ad Hoc, Snowballing</i>	R1 – Estratégias de pesquisa.
Fase 2	OE2 – Analisar técnicas e práticas de extração das métricas DORA.	A2 – Levantar técnicas possíveis.	Análise documental	R2 – Algoritmos das métricas.
		A3 – Definir algoritmos.	Lógica de primeira ordem	
Fase 3	OE3 – Definir pontos de extração dos dados para insumo das métricas.	A4 – Avaliar ferramentas existentes e dados passíveis de extração.	Revisão de ferramentas	R3 – Funcionamento e restrições do modelo.
		A5 – Entrevistar especialistas para validar extração.	Entrevista aberta	
Fase 4	OE4 – Estruturar um modelo preliminar de avaliação do desempenho <i>DevOps</i> .	A6 – Levantar requisitos funcionais e não funcionais do artefato.	Análise de requisitos	R4 – Catálogo de requisitos.
		A7 – Criar modelo arquitetural da solução.	Modelagem conceitual	R5 – Arquitetura da solução.
Fase 5	OE5 – Realizar prova de conceito (POC). ¹⁶	A8 – Desenvolver API para calcular o tempo de entrega.	Prova de conceito	R6 – Código-fonte das APIs.
		A9 – Desenvolver API para calcular a frequência de implantações.		
Fase 6	OE6 – Validar a eficácia do modelo em times <i>DevOps</i> .	A10 – Resgatar manualmente métricas de estabilidade.	Estudo de caso, entrevista exploratória e grupo focal	R7 – Avaliação do desempenho <i>DevOps</i> .
		A11 – Capturar percepções de uso em grupos focais.		
		A12 – Triangular percepções qualitativas com os dados quantitativos.		
		A13 – Avaliar e discutir resultados.		

Tabela 3.1: Metodologia em Fases

¹⁶Por motivo de restrição dos dados necessários em ferramentas proprietárias para extração das métricas de estabilidade (tempo de recuperação e taxa de falhas), estas serão recuperadas manualmente.

3.1 Procedimento de Revisão da Literatura

A revisão da literatura adotou uma abordagem adaptativa para investigar o tema *DevOps* em Instituições Financeiras. Inicialmente, foi realizada uma busca direta por publicações específicas sobre o assunto. No entanto, a escassez de estudos focados em grandes empresas do setor financeiro evidenciou a necessidade de expandir a estratégia de pesquisa, integrando metodologias complementares.

O processo foi estruturado em duas etapas principais: uma busca inicial *ad hoc*, que forneceu uma visão geral do tema, e o uso do *snowballing*, que aprofundou a análise por meio de referências adicionais. Essa combinação metodológica foi essencial para superar as limitações impostas pela disponibilidade restrita de estudos específicos na área.

3.1.1 Busca Inicial *Ad Hoc*

A *revisão ad hoc* foi realizada como a primeira etapa, adotando uma abordagem flexível para levantamento bibliográfico em bases científicas amplamente reconhecidas (GAROUSI; FELDERER; MÄNTYLÄ, 2019). Esse método pragmático visava coletar e organizar literatura relevante para estabelecer uma base inicial de conhecimento sobre práticas *DevOps* e métricas DORA.

O processo foi conduzido em três etapas principais:

- **Definição do Escopo:** A busca inicial concentrou-se em publicações relacionadas ao desempenho *DevOps* em organizações financeiras, incluindo métricas DORA e desafios culturais associados. Foram consultadas bases de dados científicas como *IEEE Xplore*, *ACM Digital Library*, *Scopus* e *Google Scholar*.
- **Seleção dos Estudos:** Foram incluídos estudos publicados entre 2013 e 2025. Critérios de exclusão aplicados visaram eliminar artigos irrelevantes ou inacessíveis. Estratégias como busca em cascata foram utilizadas para identificar trabalhos adicionais (BRERETON et al., 2007).
- **Análise dos Dados:** Os estudos selecionados foram categorizados em temas como eficácia das métricas DORA e desafios culturais no *DevOps*, complementando os dados quantitativos e qualitativos do estudo.

Quanto ao escopo e palavras-chave os termos pesquisados incluíram: "DevOps in financial institutions", "DORA metrics implementation", "DevOps culture and performance", "DevOps in large enterprises".

Incluíram-se estudos com relevância direta ao tema, disponíveis em texto completo. Excluíram-se materiais que não abordassem práticas *DevOps* ou métricas DORA no contexto organizacional.

Apesar de oferecer insights iniciais, a revisão ad hoc revelou uma lacuna significativa na literatura sobre *DevOps* no setor financeiro, justificando a adoção de uma metodologia mais robusta para aprofundar a análise.

3.1.2 Busca Complementar por *Snowballing*

Para ampliar os resultados iniciais, adotou-se a estratégia de *snowballing* como metodologia complementar, conforme descrito por Wohlin (WOHLIN, 2014). Essa abordagem iterativa permitiu explorar de forma aprofundada a literatura ao investigar tanto as referências citadas (*backward snowballing*) quanto os artigos que referenciam os estudos selecionados (*forward snowballing*).

O processo seguiu os seguintes passos:

- **Seleção do Conjunto Inicial:** Foram definidos estudos primários que atendiam aos critérios de inclusão, representando uma base inicial confiável para o *snowballing*.
- **Iteração e Expansão:** A partir do conjunto inicial, identificaram-se novos artigos relevantes por meio da análise de referências e citações, iterando até que não houvesse mais inclusão de estudos novos.
- **Validação e Análise:** Cada estudo incluído foi avaliado quanto à relevância e qualidade, garantindo sua contribuição ao enriquecimento do levantamento bibliográfico.

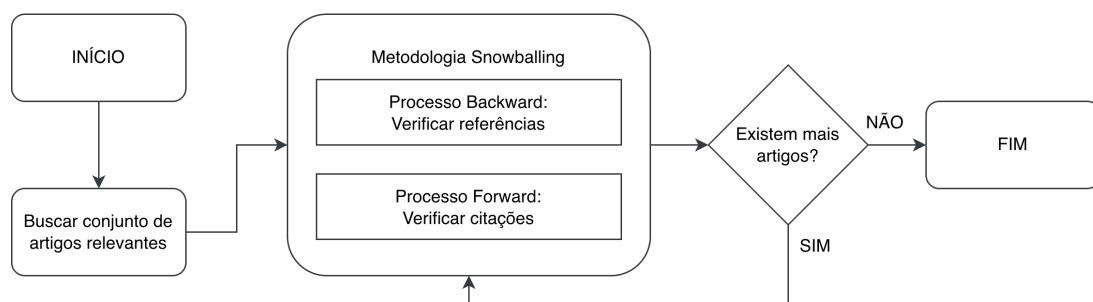


Figura 3.1: Fluxo da Metodologia de Pesquisa *Snowballing*.

O *snowballing* possibilitou a descoberta de trabalhos que não apareceram nas buscas iniciais devido à inconsistência de termos aplicados nas bases de dados (WOHLIN, 2014). Essa abordagem foi especialmente útil para identificar estudos relacionados à aplicação de métricas DORA em setores de grande porte, integrando múltiplos domínios para oferecer uma perspectiva abrangente.

Nestas iterações do *snowballing*, utilizaram-se palavras-chave complementares como: "DevOps transformation", "Agile and DevOps in banking", "Continuous delivery metrics", "DevOps performance improvement".

O processo foi conduzido em ciclos sucessivos até a saturação, garantindo que novos estudos relevantes fossem incorporados. A cada iteração, verificava-se a relevância dos artigos antes de avançar para novas referências. A estratégia iterativa permitiu combinar as descobertas da literatura com os dados coletados no estudo, criando uma base sólida para comparar diferentes perspectivas e validar os resultados de maneira mais abrangente e confiável.

3.2 Análise Documental

No entendimento de qual seria a forma mais adequada de calcular as métricas DORA dentro do contexto ferramental e cultural da instituição financeira em estudo, a análise documental assume papel central, auxiliando na identificação e evidência de pontos críticos a partir da pesquisa de documentos.

Como metodologia de investigação científica, a análise documental utiliza procedimentos técnicos e específicos para examinar e compreender o conteúdo de diferentes tipos de documentos, extraíndo deles as informações mais relevantes conforme os objetivos da pesquisa (JUNIOR et al., 2021). Independentemente do objeto de investigação, o uso de fontes documentais é imprescindível para fundamentar o estudo.

Esse procedimento recorre a métodos e técnicas voltados à apreensão, compreensão e análise de documentos variados. Os dados podem ser coletados de diferentes formas, sendo necessário definir previamente os objetivos da pesquisa para determinar a estratégia de coleta mais adequada. Busca-se, assim, identificar informações factuais em documentos a partir de questões e hipóteses de interesse, utilizando-as como objeto de estudo. A proposta metodológica pode ser aplicada tanto em abordagens qualitativas quanto quantitativas, tendo como finalidade a obtenção de informações concretas nos documentos selecionados como corpus da pesquisa (JUNIOR et al., 2021).

Segundo Godoy (1995), a análise documental, além de ser um procedimento de pesquisa com características próprias e finalidades específicas, pode também ser utilizada como técnica complementar, validando e aprofundando dados obtidos por meio de entrevistas, questionários ou observações. Para Cellard (2010), a análise consiste na obtenção de informações significativas que possibilitam a elucidação do objeto de estudo e contribuem para a solução dos problemas investigados. Esse processo envolve ainda a avaliação do contexto, dos autores, da autenticidade e confiabilidade dos documentos, da natureza do texto, dos conceitos-chave e da lógica interna apresentada.

3.3 Lógica de Primeira Ordem

As fórmulas para obtenção das métricas DORA serão padronizadas com o uso da lógica de primeira ordem (LPO), explicitando o cálculo a ser implementado.

A LPO é uma técnica formal que descreve relações, fatos e regras sobre objetos em um domínio de interesse. Ela estende a lógica proposicional ao introduzir quantificadores e predicados, permitindo expressões mais complexas sobre objetos e suas interações (VIERO, 2002).

Parte-se da premissa de que o mundo é constituído por objetos dotados de propriedades ou relações, as quais podem ser válidas ou não. Por esse motivo, a lógica de primeira ordem mantém proximidade com a linguagem natural e apresenta expressividade suficiente para formalizar toda a matemática. Sua estrutura compreende três dimensões: linguagem (símbolos e regras de formação), semântica (interpretação e significado) e axiomatização (sistemas de axiomas e demonstração de teoremas) (FAJARDO, 2017).

A LPO é composta por blocos básicos:

- **Variáveis:** símbolos que representam objetos dentro de um domínio de discurso, permitindo generalização de declarações sem especificar cada instância explicitamente (FAJARDO, 2017).
- **Quantificadores:** definem o escopo das declarações. O universal aplica-se a todas as instâncias do domínio; o existencial indica que ao menos uma instância satisfaz a afirmação (FAJARDO, 2017).
- **Predicados:** funções que retornam valores booleanos e descrevem propriedades ou relações entre objetos, possibilitando expressões complexas (FAJARDO, 2017).
- **Funções:** mapeiam tuplas de objetos para outros objetos, representando relacionamentos que produzem saídas específicas a partir de entradas (FAJARDO, 2017).
- **Constantes:** representam objetos específicos e imutáveis do domínio, permitindo declarações sobre entidades particulares (FAJARDO, 2017).

Dessa forma, a lógica de primeira ordem foi utilizada como técnica formal para estruturar e extrair os resultados das fórmulas que fundamentam o cálculo das métricas DORA, conforme apresentado na Seção 4.2.

3.4 Revisão de Ferramentas

Insumos mínimos são necessários para o funcionamento adequado das métricas DORA. A revisão das ferramentas *DevOps* utilizadas na instituição financeira em estudo tem como objetivo analisar a possibilidade de extração de dados, evidenciando quais são as permissões e restrições de acesso existentes.

Para conhecer as ferramentas disponíveis, será realizada a navegação pelas telas e funcionalidades, conferindo as informações acessíveis. Os dados necessários para viabilizar os cálculos das métricas DORA, resultantes da atividade de definição dos algoritmos (A3), serão organizados como uma lista de critérios. Durante a execução do fluxo de navegação, as funcionalidades serão avaliadas com o propósito de verificar se algum dado é relevante conforme os critérios estabelecidos. Após esse levantamento, caso sejam identificados dados pertinentes, a equipe responsável pela ferramenta analisará a possibilidade de disponibilização para utilização nos algoritmos.

Quanto à avaliação da qualidade dos dados permitidos para uso, será conduzida uma revisão sistemática baseada na análise comparativa entre os resultados obtidos e a fidelidade das informações, considerando sua aplicabilidade prática.

3.5 Entrevista Aberta

Para a validação dos resultados dos dados viáveis que servirão de insumo ao cálculo das métricas DORA, com o propósito de alinhamento entre as áreas e engajamento da solução para futuro uso corporativo, entrevistas aos especialistas serão conduzidas. A ideia é embasar informações para garantir a qualidade das funcionalidades que serão desenvolvidas, além de evidenciar o potencial dos resultados deste trabalho.

A entrevista aberta trata-se de técnica de extração de informações por meio de conversa entre duas ou mais pessoas. O entrevistador tem a responsabilidade de conduzir uma investigação sobre um assunto específico. Espera-se que o produto do encontro resulte em dados concretos sobre o que precisa ser evidenciado. Na entrevista aberta o entrevistador é um facilitador e atua de maneira imparcial, deixando o entrevistado livre em sua manifestação com o propósito de permitir uma imagem rica, profunda e contextualizada (RAPLEY, 2001).

Embora esta técnica seja direcionada à entrevista estruturada¹⁷, a maioria dos praticantes da entrevista aberta rejeita a ideia de produzir dados quantificáveis fixos. O tema é conduzido de forma dinâmica e interpretado a partir de sentimentos reais, ideias e

¹⁷Tipo de entrevista voltada tanto para objetivos quantitativos quanto qualitativos, com regras bem definidas e um roteiro fixo de perguntas.

emoções pessoais. Por meio do método simples de formular questões iniciais e de acompanhamento, durante a entrevista o que permanece relevante é reforçado pelo entrevistador em sua interação (RAPLEY, 2001).

Entrevistadores devem possuir controle abrangente do tópico, guiando a conversa e promovendo-a por meio de perguntas, silêncio e anotações das respostas, considerando o que possui maior importância para evidências. A qualidade da entrevista é influenciada pela postura analítica escolhida sobre os dados, sobre a consciência e sensibilidade à forma como os entrevistados e entrevistadores produzem colaborativamente a conversa. Trajetórias alternativas muitas vezes silenciosas, de reflexão podem ser oportunas e positivas ao processo a depender da sensibilidade das partes (RAPLEY, 2001).

Ao considerar o andamento da entrevista, o entrevistador deve se manter ativo na interação. Como tal, a conversa deve ser produzida num contexto específico e a consciência desse contexto é vital para a compreensão da conversa e, portanto, dos próprios dados. No que diz respeito às práticas dos entrevistadores, qualquer que seja o ideal metodológico sobre as práticas que são produzidas, seja facilitação neutra ou auto-revelação cooperativa, nenhum ideal único obtém melhores dados do que os outros. Qualquer que seja a postura analítica adotada, não se pode escapar da natureza interativa das entrevistas, de que os dados são produzidos de forma colaborativa. Qualquer que seja o ideal que o entrevistador pratique, a sua fala é central para a trajetória da fala do entrevistado. Como tal, os dados obtidos na entrevista específica começam a emergir como apenas uma versão possível, uma versão que depende do contexto interativo específico (RAPLEY, 2001).

3.5.1 Entrevista Exploratória

Para complementar a análise quantitativa e qualitativa das métricas DORA, adotou-se a técnica de entrevista exploratória como método de coleta de informações. Essa abordagem busca aprofundar o entendimento sobre os desafios enfrentados pelos times *DevOps* e as percepções relacionadas à aplicação das métricas, além de promover o engajamento das áreas envolvidas na solução proposta. As entrevistas serão conduzidas com especialistas e profissionais de diferentes setores, com o objetivo de validar os dados utilizados no cálculo das métricas, alinhar expectativas e fortalecer a aplicabilidade corporativa do modelo desenvolvido.

Esse tipo de entrevista caracteriza-se pela flexibilidade e adaptabilidade de suas questões, permitindo ao entrevistador investigar temas emergentes durante a conversa. Diferentemente das entrevistas estruturadas, procura captar percepções ricas e contextuais, fundamentadas nas experiências e interpretações dos participantes (RAPLEY, 2001). O entrevistador atua como facilitador, promovendo um ambiente de diálogo aberto e orientando a discussão para explorar aspectos relevantes ao estudo.

Os principais objetivos dessa metodologia incluem:

- Identificar barreiras culturais e operacionais para adoção das métricas DORA.
- Explorar junto aos especialistas a eficácia das métricas no contexto corporativo.
- Validar os dados e os métodos de cálculo utilizados no modelo, garantindo maior alinhamento com a realidade das operações.
- Coletar sugestões para melhorias nas práticas de adoção das métricas e na evolução do modelo.

Estratégia de Condução

Para garantir a eficácia da coleta de informações, serão seguidas as seguintes etapas:

- **Planejamento:** Seleção criteriosa dos participantes, priorizando especialistas e demais funções (desenvolvedores, gerentes, operações) em diferentes setores, para garantir diversidade de perspectivas (STEWART; SHAMDASANI, 2014).
- **Elaboração de roteiro:** Desenvolvimento de questões abertas e flexíveis, que permitam explorar percepções e desafios enfrentados. Exemplos de questões iniciais incluem:
 - Como você avalia a aplicação das métricas DORA no contexto de sua equipe?
 - Quais dificuldades sua área enfrenta na utilização dessas métricas?
 - Que melhorias você sugeriria para ampliar a efetividade das métricas?
- **Condução da entrevista:** O entrevistador incentivará a livre manifestação dos participantes, ao mesmo tempo em que guiará a conversa com base nos objetivos definidos, utilizando perguntas de acompanhamento e silêncios estratégicos para aprofundar as discussões (RAPLEY, 2001).
- **Registro e análise:** As respostas serão registradas por meio de anotações detalhadas ou gravações (mediante consentimento), e posteriormente analisadas para identificar padrões e temas recorrentes.

A entrevista exploratória foi escolhida devido à sua capacidade de gerar *insights* detalhados e específicos sobre as percepções de cada participante, alinhando-se aos objetivos do estudo. Essa abordagem possibilita a coleta de dados ricos e contextuais, que complementam a análise quantitativa das métricas DORA e fornecem subsídios para futuras ações de melhoria (RAPLEY, 2001).

Além disso, ao promover o diálogo entre as áreas envolvidas, essa técnica contribui para a construção de um entendimento compartilhado sobre o modelo proposto, fortalecendo sua aplicabilidade e engajamento em escala corporativa.

Embora seja uma metodologia para compreender contextos específicos, a entrevista exploratória possui limitações, como a dependência da habilidade do entrevistador e a possibilidade de viés na interpretação dos dados. Ainda assim, quando conduzida de forma criteriosa, oferece valiosas contribuições qualitativas que ampliam a compreensão dos fenômenos analisados (RAPLEY, 2001).

3.5.2 Grupo Focal

O *grupo focal* será utilizado como metodologia qualitativa complementar à análise quantitativa das métricas DORA. Trata-se de uma técnica de coleta de dados qualitativos que consiste em reunir um pequeno grupo de participantes para discutir um tema específico sob a orientação de um moderador. O objetivo é captar percepções, opiniões e experiências dos participantes, promovendo interação e troca de ideias (MORGAN, 1997).

- Explorar como os participantes percebem a eficácia das métricas DORA no contexto de suas atividades diárias.
- Identificar mudanças culturais e comportamentais geradas pela adoção das métricas DORA, como resistências ou avanços em relação às práticas ágeis.
- Obter feedback sobre os principais desafios enfrentados pelos times e sugerir melhorias nas práticas e na utilização das métricas (KRUEGER; CASEY, 2014).

Estrutura do Grupo Focal

- **Participantes:** Membros de diferentes funções dentro dos times *DevOps* (desenvolvedores, gerentes, operações) (STEWART; SHAMDASANI, 2014).
- **Moderador:** O moderador será responsável por facilitar a discussão, assegurar que os temas relevantes sejam abordados e garantir que todos os participantes tenham oportunidade de expressar suas opiniões. (GIBBS, 2012).
- **Tamanho:** 6-10 participantes por grupo, proporcionando diversidade de opiniões sem comprometer a profundidade da discussão (MORGAN, 1997).
- **Duração:** Cada sessão de grupo focal terá duração de aproximadamente 60 a 90 minutos, tempo suficiente para cobrir os tópicos relevantes sem perder o foco e engajamento dos participantes (KRUEGER; CASEY, 2014).

Questões Orientadoras

As questões orientadoras do grupo focal devem ser elaboradas de forma a incentivar discussões profundas e abertas. Algumas questões iniciais podem incluir:

- Como as métricas DORA têm influenciado o seu trabalho diário e as interações dentro do time?
- Quais barreiras culturais você identificou na adoção das métricas DORA?
- Você acredita que as métricas DORA ajudam a melhorar o desempenho da equipe? Se sim, de que maneira?
- Quais desafios você enfrentou ao tentar aplicar as métricas DORA em sua equipe?
- Quais mudanças você notou nas práticas de *DevOps* desde a introdução das métricas DORA?

Essas perguntas serão abertas, permitindo que os participantes compartilhem suas experiências e opiniões de forma livre. O moderador também poderá adaptar as perguntas com base nas respostas dos participantes, promovendo discussões ricas e específicas (KRUEGER; CASEY, 2014).

Estratégia para Uso das Questões

Para garantir que o grupo focal seja uma ferramenta eficaz de coleta de dados, as questões orientadoras podem ser ajustadas conforme o grupo de participantes. Por exemplo:

- Para grupos compostos majoritariamente por desenvolvedores, a ênfase será dada ao impacto das métricas DORA no ciclo de desenvolvimento e entrega de software.
- Em grupos compostos por gerentes ou operadores, a abordagem será mais voltada para os efeitos das métricas DORA nas decisões estratégicas e na eficiência operacional.

Essa abordagem flexível permite que o moderador adapte as questões de acordo com as funções e experiências dos participantes, garantindo que a discussão se mantenha relevante e focada em aspectos específicos de cada grupo (TURNER; MCALLISTER, 2017).

Justificativa

O grupo focal é ideal para este estudo, pois possibilita a coleta de *insights* detalhados sobre as percepções de diferentes *stakeholders*. Ao gerar discussões interativas, ele oferece uma

rica perspectiva sobre o antes e depois da adoção das métricas DORA, complementando os dados quantitativos obtidos com as métricas (KRUEGER; CASEY, 2014).

Além disso, permite que se obtenham informações qualitativas sobre a aplicação prática das métricas e os impactos reais no comportamento dos times, fornecendo dados valiosos para a análise de eficácia das métricas e o desempenho *DevOps* no contexto do setor financeiro. Esta metodologia permite a triangulação entre os resultados das métricas, entrevistas individuais e discussões de grupo, ampliando a compreensão sobre a maturidade *DevOps* no contexto do setor financeiro (STEWART; SHAMDASANI, 2014).

3.6 Análise de Requisitos

A atividade de levantamento de requisitos funcionais e não funcionais (A6) foi conduzida como parte da Fase 4 da metodologia, utilizando a técnica de análise de requisitos. O objetivo dessa etapa é explicitar as necessidades que o artefato deve atender antes da modelagem arquitetural, garantindo que os diagramas e modelos propostos estejam fundamentados em requisitos claros e verificáveis.

Procedimentos Metodológicos

A análise de requisitos foi realizada por meio de:

- **Entrevistas exploratórias** com especialistas da instituição financeira, para identificar necessidades práticas e restrições operacionais.
- **Revisão documental** das ferramentas utilizadas (RTC, GitLab, Jenkins, Sonar, JFrog, GSTI), com foco em suas capacidades e limitações técnicas.
- **Mapeamento das métricas DORA** como requisitos funcionais centrais, assegurando que o artefato suporte o cálculo de frequência de implantações, tempo de entrega, tempo médio de recuperação e taxa de falhas.
- **Classificação dos requisitos** em funcionais e não funcionais, conforme práticas consolidadas de engenharia de software.

Integração com a Metodologia

Essa etapa metodológica não se limita à coleta de requisitos, mas estabelece a ponte entre os insumos levantados nas fases anteriores (dados e entrevistas) e a modelagem arquitetural da solução (A7). A análise de requisitos garante que:

- Os diagramas arquiteturais sejam derivados de necessidades explícitas.

- As métricas DORA sejam incorporadas como funcionalidades obrigatórias.
- As restrições de desempenho, segurança e escalabilidade sejam consideradas desde o início da modelagem.

Análise de Requisitos Não Funcionais com URPS+

Para a identificação dos requisitos não funcionais, foi utilizada a metodologia URPS+, que organiza os requisitos em quatro categorias principais — Usabilidade, Confiabilidade, Performance e Suportabilidade — acrescidas de atributos adicionais como segurança e interoperabilidade. Essa abordagem é reconhecida na literatura de engenharia de software como uma técnica sistemática para elicitación de requisitos não funcionais (CYSNEIROS, 2001; BASS; CLEMENTS; KAZMAN, 2012).

- **Usabilidade (U)** – facilidade de uso da solução, acessibilidade e clareza das interfaces.
- **Confiabilidade (R)** – disponibilidade contínua, tolerância a falhas e consistência dos resultados.
- **Performance (P)** – tempo de resposta adequado, eficiência no processamento das métricas DORA.
- **Suportabilidade (S)** – escalabilidade para múltiplos times, facilidade de manutenção e compatibilidade com ambientes de nuvem.
- **Atributos adicionais (+)** – segurança dos dados, conformidade regulatória e interoperabilidade com ferramentas de CI/CD.

A aplicação da metodologia URPS+ permitiu consolidar os requisitos não funcionais em um catálogo estruturado, que complementa os requisitos funcionais e orienta a modelagem arquitetural da solução.

3.7 Modelagem Conceitual

Neste método, os participantes criam um modelo de algum aspecto de seu trabalho, com a intenção de explicitar seus modelos mentais. Na forma mais simples, os participantes desenham um diagrama relacionado às suas necessidades. Por exemplo, engenheiros de *software* podem ser solicitados a elaborar um diagrama de fluxo de dados, um diagrama de fluxo de controle ou um diagrama de pacotes que represente os *clusters* arquitetônicos do sistema.

Os modelos conceituais fornecem um retrato preciso da concepção do usuário sobre seu modelo mental do sistema. Tais modelos são de fácil elaboração e favorecem o entendimento sobre o que precisa ser construído. Para modelar as APIs das métricas DORA, será utilizada a linguagem de modelagem unificada (UML), evidenciando responsabilidades, comunicação entre camadas, bem como classes e componentes específicos.

3.7.1 Linguagem de Modelagem Unificada (UML)

A UML, ou *Unified Modeling Language*, é uma linguagem de notação destinada à modelagem e documentação das fases de desenvolvimento de softwares orientados a objetos. Seu papel é auxiliar desenvolvedores a visualizar os diversos aspectos da aplicação, facilitando a compreensão do funcionamento do sistema (BOOCH, 2006).

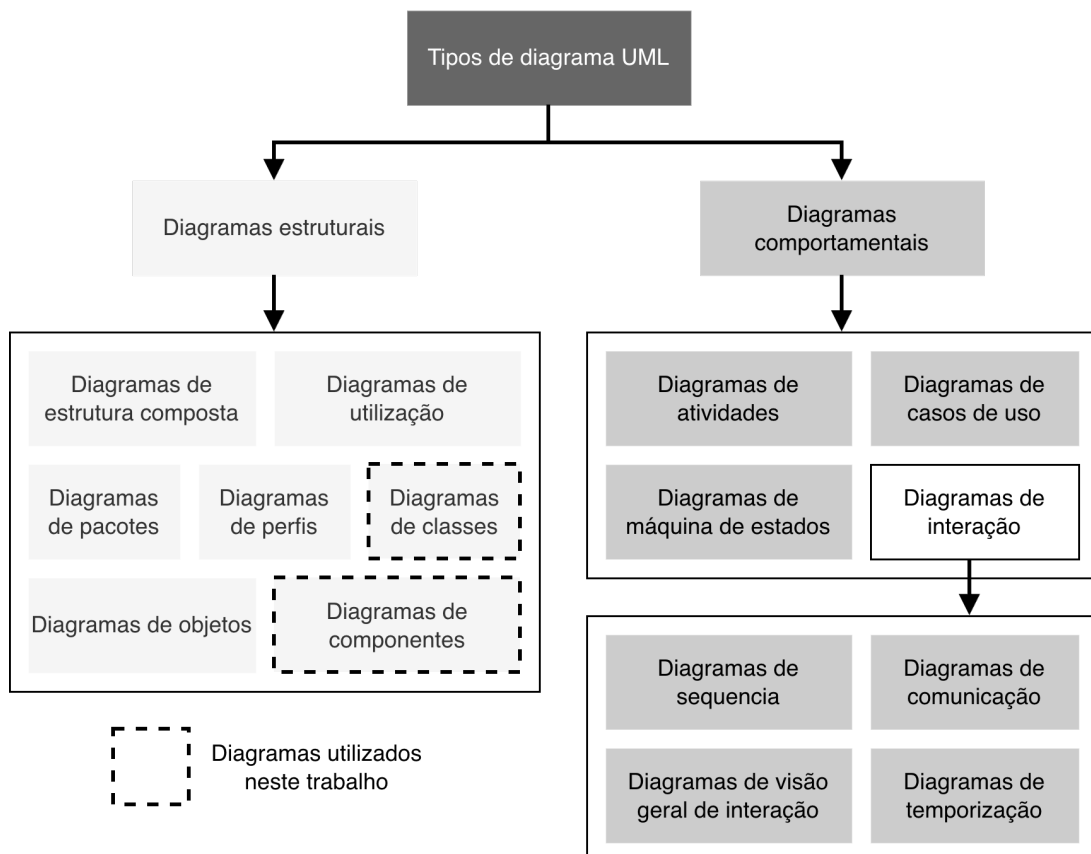


Figura 3.2: Tipos de diagramas da UML.

A linguagem UML é composta por quatorze diagramas, organizados em duas categorias: estruturais e de comportamento. Os diagramas estruturais representam o aspecto estático do sistema e das classes, abrangendo dois dos quatro pontos de vista da modelagem. Os diagramas de comportamento descrevem o sistema em execução, ou seja, a modelagem dinâmica. Essa abordagem contempla tanto a visão dinâmica do sistema

quanto a das classes, completando as quatro perspectivas necessárias para uma especificação abrangente (JACOBSON; BOOCH; RUMBAUGH, 2021).

No escopo deste trabalho, são apresentados apenas os diagramas utilizados, destacados na Figura 3.2, que sustentam a construção arquitetural da solução detalhada no Capítulo 4 de resultados.

Diagrama de Classes

O diagrama de classes é o modelo fundamental de uma especificação orientada a objetos (JACOBSON; BOOCH; RUMBAUGH, 2021). Ele é utilizado tanto para a modelagem conceitual da estrutura geral da aplicação quanto para a modelagem detalhada, permitindo a tradução dos modelos em código de programação. Além disso, pode ser empregado na modelagem de dados. As classes representadas nesse diagrama descrevem os principais elementos do sistema, suas interações e as classes que deverão ser implementadas (BOOCH, 2006).

Esse tipo de diagrama fornece a descrição mais próxima da estrutura do código de um programa, evidenciando o conjunto de classes com seus atributos e métodos, bem como os relacionamentos entre elas. Classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (JACOBSON; BOOCH; RUMBAUGH, 2021).

As classes são utilizadas para capturar o vocabulário do sistema em desenvolvimento. Podem incluir abstrações pertencentes ao domínio do problema, assim como classes que compõem a implementação. Dessa forma, representam itens de *software*, itens de *hardware* ou até mesmo elementos puramente conceituais (BOOCH, 2006).

Diagrama de Componentes

O diagrama de componentes tem como finalidade especificar os elementos de um sistema e seus relacionamentos. Um componente é uma parte substituível que realiza um conjunto de interfaces e pode residir em um nó, como executáveis, bibliotecas, tabelas, arquivos ou documentos. Normalmente, representa o empacotamento físico de elementos lógicos, como classes, interfaces e colaborações (JACOBSON; BOOCH; RUMBAUGH, 2021).

Na UML, todos os itens físicos são modelados como componentes, conectando modelos lógicos e físicos por meio de interfaces. Por exemplo, uma interface definida em um modelo lógico pode ser realizada por um componente físico correspondente (JACOBSON; BOOCH; RUMBAUGH, 2021).

Esses diagramas permitem verificar se a funcionalidade necessária de um sistema é atendida e servem como ferramenta de comunicação entre desenvolvedores e partes interessadas (BOOCH, 2006). Para os desenvolvedores, formalizam um roteiro de implementação e auxiliam na atribuição de tarefas. Para administradores de sistema, oferecem

uma visão antecipada dos componentes de *software* e seus relacionamentos, apoiando o planejamento e a gestão da infraestrutura (BOOCH, 2006).

3.8 Prova de Conceito

A execução de uma prova de conceito (*proof-of-concept* – PoC) constitui um passo essencial para o sucesso de iniciativas em larga escala. Essa prática permite identificar problemas antecipadamente, minimizar riscos na implementação definitiva e, em muitos casos, reduzir custos (ALUNNI, 2019). Além disso, favorece o engajamento e a confiança das partes interessadas, ao possibilitar validar a ideia em menor escala antes de sua adoção ampla.

No contexto da mensuração do desempenho *DevOps* em uma instituição financeira, a PoC representa um instrumento fundamental para sustentar a evolução tecnológica. Por meio dela, é possível coletar dados relevantes, experimentar funcionalidades específicas e gerar aprendizado sobre pontos fortes e fragilidades, fornecendo insumos concretos para decisões estratégicas.

A PoC é, portanto, um processo voltado a demonstrar a viabilidade de uma solução e sua aplicação prática (ALUNNI, 2019). No âmbito do *software*, pode ser utilizada para avaliar novas tecnologias destinadas à automação de tarefas críticas ou validar processos específicos dentro da organização. Esse tipo de experimento permite identificar pontos problemáticos e desenvolver práticas recomendadas sem expor o ambiente real a grandes riscos.

Para viabilizar o desenvolvimento das APIs responsáveis pela consulta das métricas DORA, serão aplicadas técnicas de engenharia de *software*. Pretende-se construir essas APIs considerando a arquitetura MVC e os princípios da programação orientada a objetos, detalhados nas seções seguintes.

3.8.1 Arquitetura *Model-View-Controller* (MVC)

O padrão arquitetural *Model-View-Controller* (MVC) é amplamente consolidado na engenharia de *software*, sendo adotado em diversos frameworks modernos como Spring, Django e Ruby on Rails (KUNJUMOHAMED; SATTARI, 2016). Sua principal característica é a separação de responsabilidades em três componentes independentes: *Model*, responsável pela lógica de negócio e persistência de dados; *View*, encarregada da apresentação das informações ao usuário; e *Controller*, que atua como intermediário, recebendo entradas e coordenando a interação entre modelo e visualização. Essa divisão favorece a modularidade e reduz o acoplamento entre camadas, o que se traduz em maior facilidade de manutenção e evolução do sistema (GAMMA et al., 1995).

A Figura 3.3 ilustra essa dinâmica, evidenciando o fluxo de interação entre os componentes. O *Controller* recebe uma ação ou evento do usuário, atualiza o *Model*, que por sua vez notifica o *Controller*, e este atualiza a *View*. A *View* expõe o modelo, permitindo que o usuário visualize os dados atualizados.

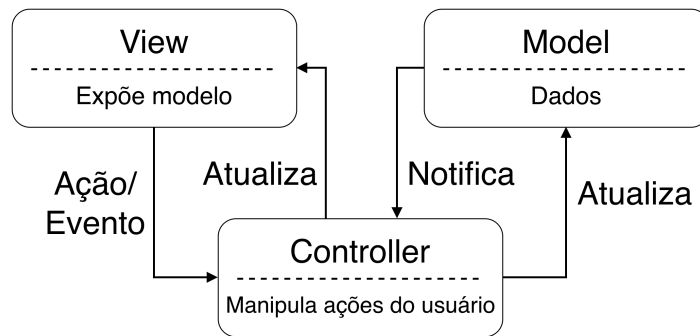


Figura 3.3: Fluxo de interação entre os componentes da arquitetura MVC.

Do ponto de vista técnico, o MVC apresenta vantagens claras em relação à arquitetura em camadas tradicional. Embora esta seja amplamente utilizada, tende a gerar forte acoplamento entre lógica de negócio e apresentação, dificultando a manutenção em sistemas de larga escala (FOWLER, 2002). O MVC surgiu como uma evolução desse modelo, propondo uma separação mais clara de responsabilidades entre dados, lógica de controle e apresentação. No contexto de APIs, a camada de apresentação é representada pela resposta serializada (como JSON ou XML), o que mantém a aplicabilidade do padrão sem comprometer sua utilidade. Essa característica garante que alterações na interface ou nos dados possam ser realizadas de forma independente, sem comprometer o restante da aplicação. Em ambientes corporativos complexos, como instituições financeiras, essa flexibilidade é particularmente relevante, pois confiabilidade e rastreabilidade são requisitos críticos.

Outra alternativa frequentemente considerada é o padrão MVVM¹⁸. Embora ofereça vantagens em cenários específicos, o MVVM não apresenta a mesma clareza de separação em aplicações orientadas a serviços, como APIs voltadas à mensuração de métricas DORA.

Entre as alternativas discutidas na literatura recente, destaca-se também a arquitetura hexagonal¹⁹. Essa arquitetura é frequentemente citada como opção para sistemas que

¹⁸O padrão MVVM (*Model-View-ViewModel*) foi introduzido por John Gossman em 2005 como uma variação do MVC, inicialmente voltado para aplicações *desktop* no framework WPF da Microsoft. Sua principal inovação é a camada *ViewModel*, que atua como intermediária entre *View* e *Model*, permitindo maior desacoplamento e facilitando o uso de *data binding*. Essa abordagem é especialmente útil em interfaces gráficas complexas, mas apresenta menor aplicabilidade em arquiteturas orientadas a serviços e APIs.

¹⁹A arquitetura hexagonal, também conhecida como *Ports and Adapters*, foi proposta por Alistair Cockburn. Seu objetivo é isolar o núcleo da aplicação das dependências externas, como bancos de

exigem alto grau de independência entre o domínio e suas interfaces externas. Embora promova flexibilidade e isolamento de dependências, exige maior esforço de modelagem e complexidade inicial, o que poderia comprometer a agilidade necessária ao experimento.

Portanto, a adoção do MVC nesta prova de conceito não se deve a conveniência ou familiaridade individual, mas à ampla utilização da arquitetura dentro da instituição financeira em estudo. O MVC é empregado em diversos sistemas internos, o que assegura maior aderência às práticas existentes, reduz a curva de aprendizado dos times e facilita a integração com soluções previamente estabelecidas. Essa decisão metodológica está alinhada às boas práticas de engenharia de *software* e fundamentada em literatura consolidada (GAMMA et al., 1995; FOWLER, 2002), garantindo que a arquitetura escolhida seja adequada ao objetivo de disponibilizar APIs para consulta das métricas DORA em um ambiente financeiro regulado.

3.8.2 Programação Orientada a Objetos

A programação, assim como diversas atividades cotidianas, pode ser organizada segundo diferentes paradigmas. Entre esses paradigmas, destaca-se a programação orientada a objetos (POO) (WIENER; PINSON, 2000). Esse paradigma constitui uma abordagem para a solução de problemas em que os cálculos são realizados no contexto de objetos. Os objetos correspondem a instâncias de construções de programação, geralmente denominadas classes, que representam abstrações de dados e incluem procedimentos responsáveis por operar sobre esses objetos (LETHBRIDGE; LAGANIERE, 2005).

A programação orientada a objetos configura-se, portanto, como um método de organização e construção de *software*, além de uma forma de estruturar a resolução de problemas (WIENER; PINSON, 2000). Esse paradigma surgiu como alternativa às características da programação estruturada, com o propósito de aproximar o tratamento das estruturas de um programa ao tratamento das entidades do mundo real. O termo “objeto” foi adotado para designar um elemento genérico capaz de representar qualquer entidade tangível (WIENER; PINSON, 2000).

Um objeto de *software* representa uma abstração de determinada realidade. Essa realidade pode corresponder a um objeto físico, mas é mais frequentemente uma ideia ou conceito representado por um estado interno (WIENER; PINSON, 2000). A POO também se fundamenta na noção de envio de mensagens a objetos. Essas mensagens podem modificar ou retornar informações sobre o estado interno de um objeto. O comportamento de um objeto é codificado em uma descrição de classe, sendo o objeto considerado uma

dados, interfaces gráficas ou serviços de terceiros. Essa abordagem promove flexibilidade e facilita a substituição de componentes externos sem impactar a lógica de negócio, sendo especialmente útil em sistemas complexos e de longa manutenção.

instância da classe que define seu comportamento. A descrição da classe especifica o estado interno do objeto e estabelece os tipos de mensagens que podem ser enviadas para todas as suas instâncias (WIENER; PINSON, 2000).

Esse paradigma fundamenta-se principalmente em dois conceitos-chave: classes e objetos. Todos os demais conceitos são derivados desses elementos (WIENER; PINSON, 2000). De forma análoga a um molde, a classe corresponde a um conjunto de características e comportamentos que definem os objetos pertencentes a ela. A classe, por si só, é um conceito abstrato, que se torna concreto por meio da criação de um objeto. Esse processo é denominado instanciação da classe, em que o molde é utilizado para gerar um objeto (WIENER; PINSON, 2000). Classes e objetos constituem, assim, os pilares da programação orientada a objetos.

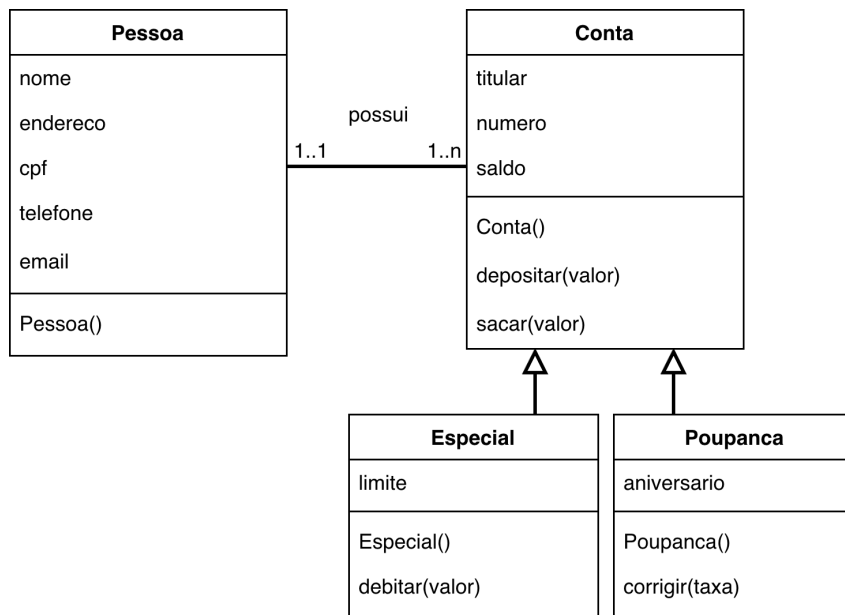


Figura 3.4: Exemplo de Diagrama de Classes representando Contas Bancárias.

- **Objetos:** bloco de dados estruturados em um sistema de *software* em execução. Pode representar qualquer coisa com a qual você possa associar propriedades e comportamento. As propriedades caracterizam o objeto, descrevendo seu estado atual. Comportamento é a forma como um objeto age e reage, possivelmente mudando seu estado (LETHBRIDGE; LAGANIERE, 2005).
- **Classes:** unidades de abstração de dados num programa orientado a objetos. Mais especificamente, uma classe é um módulo de *software* que representa e define um conjunto de objetos semelhantes, suas instâncias. Todos os objetos com as mesmas propriedades e comportamento são instâncias de uma classe (LETHBRIDGE; LAGANIERE, 2005).

O código descreve como os objetos da classe são estruturados, ou seja, os dados armazenados em cada objeto que implementam as propriedades. Os procedimentos, chamados métodos, que implementam o comportamento dos objetos (LETHBRIDGE; LAGANIERE, 2005).

3.9 Estudo de Caso

A validade de um estudo de caso deve ser considerada desde o início da pesquisa. Antes da definição do tema deste trabalho, foram investigados exemplos e estratégias voltadas à melhoria da qualidade no desenvolvimento e na manutenção de *software*, com base em protocolos metodológicos detalhados. Uma vez estabelecido o objeto de estudo e mobilizados os recursos necessários, a coleta de dados e a análise dos resultados serão conduzidas junto aos sujeitos envolvidos. Também será contemplada a busca por teorias que contradigam as descobertas, isto é, a análise de casos negativos, com o objetivo de compreender possíveis impactos decorrentes da aplicação das métricas DORA.

O propósito de um estudo de caso é esclarecer uma “classe” de fenômenos. Procura-se construir confiança em evidências extraídas de um único caso e, ao mesmo tempo, iluminar características de um conjunto mais amplo de situações (YIN, 2009). Essa abordagem pode ser utilizada tanto em estratégias de caráter causal ou exploratório quanto descritivo. O estudo de caso representa uma investigação empírica abrangente, que envolve planejamento, coleta e análise de dados. Pode incluir estudos de caso único ou múltiplos, bem como combinar abordagens quantitativas e qualitativas (YIN, 2009).

A classificação pode ser intrínseca ou particular, quando se busca compreender um caso em seus próprios aspectos; instrumental, quando o caso é examinado para esclarecer uma questão mais ampla ou orientar pesquisas futuras; e coletiva, quando se estende a análise a múltiplos casos instrumentais relacionados, ampliando a compreensão ou a teorização sobre um conjunto maior de fenômenos (DENZIN; LINCOLN, 2011).

As chances de sucesso da pesquisa baseada em estudo de caso aumentam significativamente quando há um *design* bem definido. Isso inclui a formulação de uma pergunta inicial clara e objetiva, proposições orientadoras, definição das unidades de análise e critérios de interpretação dos achados, todos alinhados ao referencial teórico (YIN, 2009). Gomes (GOMES, 2008) destaca que, como a coleta e a análise de dados ocorrem de forma concomitante, o pesquisador deve lidar com evidências convergentes e inferências, exigindo capacidade interpretativa para compreender as respostas obtidas. Em outras palavras, “boas respostas” dependem de “boas perguntas”. O autor recomenda ainda alguns cuidados fundamentais para assegurar a qualidade dos dados coletados em campo, tais como:

- antecipar contatos com possíveis entrevistados ou gestores das organizações que serão sujeitos da pesquisa;
- organizar previamente o material necessário (equipamentos de gravação, instrumentos para anotações etc.);
- estabelecer um cronograma detalhado para a pesquisa de campo;
- sempre que possível, realizar estudos de caso-piloto, capazes de antecipar situações que poderão ser enfrentadas durante a investigação.

3.9.1 Componentes do Estudo

Questões de Estudo

A pesquisa de estudo de caso é do tipo flexível, conforme mencionado anteriormente. Isso não significa que o planejamento seja desnecessário. Pelo contrário, um bom planejamento de um estudo de caso é crucial para o seu sucesso. Há várias questões de estudo que precisam de ser planejadas, tais como quais os métodos a utilizar para a recolha de dados, que departamentos de uma organização visita, que documentos são lidos, que pessoas serão entrevistadas, com que frequência as entrevistas devem ser realizadas, e outras (RUNESON; HöST, 2009).

O objetivo do estudo pode ser, por exemplo, exploratório, descritivo, explicativo ou de melhoria. O objetivo é naturalmente formulado de forma mais geral e menos preciso do que em projetos de pesquisa fixos. O objetivo inicialmente é mais como um ponto focal que evolui durante o estudo. As questões de estudo estabelecem o que é necessário saber para cumprir o objetivo. Semelhante ao objetivo, as questões de estudo evoluem durante o trabalho e são reduzidas a questões de estudo específicas durante as iterações (RUNESON; HöST, 2009).

Preparação para a Coleta de Dados

Sobre as técnicas de coleta, Lethbridge et al. (LETHBRIDGE; SIM; SINGER, 2005) relata métodos classificados em graus:

- Primeiro grau: Métodos diretos significa que o pesquisador está em contato direto com os sujeitos e coleta dados em tempo real. Este é o caso, por exemplo, de entrevistas, grupos focais e observações (LETHBRIDGE; SIM; SINGER, 2005).
- Segundo grau: Métodos indiretos em que o pesquisador coleta dados brutos diretamente, sem realmente interagir com os sujeitos durante a coleta de dados. Esta abordagem é adotada, por exemplo, na Telemetria de Projetos de *software*, onde

o uso de ferramentas de engenharia de *software* é monitorado automaticamente e observado (LETHBRIDGE; SIM; SINGER, 2005).

- Terceiro grau: Análise independente de artefatos de trabalho em que são utilizados dados disponíveis e, em alguns casos, compilados. Esse tipo de análise ocorre, por exemplo, quando documentos como especificações de requisitos e relatórios de falhas de uma organização são examinados ou quando dados de bancos de dados organizacionais, como registros de contabilidade de tempo, são avaliados (LETHBRIDGE; SIM; SINGER, 2005).

Os métodos de primeiro grau são, em sua maioria, mais caros de aplicar do que os métodos de segundo ou terceiro grau, uma vez que exigem um esforço significativo tanto do pesquisador quanto dos sujeitos (LETHBRIDGE; SIM; SINGER, 2005).

Critérios de Qualidade

De acordo com Easterbrook et al. (EASTERBROOK et al., 2008), para se derivar as proposições de estudo de uma teoria, deve-se projetar o estudo para abordar as proposições e, em seguida, tirar conclusões mais gerais dos resultados. Cada uma dessas etapas deve ser comprovada como correta para garantir a qualidade dos critérios.

- **Validade de construto:** Este aspecto da validade reflete até que ponto as medidas operacionais estudadas realmente representam o que o pesquisador tem em mente e o que é investigado de acordo com as questões de pesquisa (RUNESON; HÖST, 2009).
- **Validade interna:** Este aspecto da validade é preocupante quando as relações causais são examinadas (RUNESON; HÖST, 2009).
- **Validade externa:** Este aspecto da validade preocupa-se com até que ponto é possível generalizar os resultados e até que ponto os resultados são de interesse para outras pessoas fora do caso investigado. Durante a análise da validade externa, o pesquisador tenta analisar até que ponto os resultados são relevantes para outros casos. Para estudos de caso, a intenção é permitir a generalização analítica onde os resultados são estendidos a casos que têm características comuns e, portanto, para os quais os resultados são relevantes, ou seja, definindo uma teoria (RUNESON; HÖST, 2009).
- **Confiabilidade:** Este aspecto diz respeito a até que ponto os dados e a análise dependem dos investigadores específicos. Hipoteticamente, se outro pesquisador conduzisse posteriormente o mesmo estudo, o resultado deveria ser o mesmo.

Ameaças na Validade

Quando o pesquisador está investigando se um fator afeta um fator investigado, existe o risco de o fator investigado também ser afetado por um terceiro fator. Se o pesquisador não tiver conhecimento do terceiro fator ou não souber até que ponto ele afeta o fator investigado, existe uma ameaça à validade interna (RUNESON; HöST, 2009). As ameaças ao aspecto da validade são, por exemplo, se não estiver claro como codificar os dados recolhidos ou se os questionários ou as perguntas das entrevistas não forem claros (RUNESON; HöST, 2009)

A validade do estudo de caso deve ser considerada desde o início. Exemplos de estratégias para aprimorar essa validade incluem a triangulação, o desenvolvimento e a manutenção de um protocolo detalhado, a revisão de projetos e protocolos por pesquisadores pares, a coleta de dados e a obtenção de resultados revisados pelos sujeitos do caso, a dedicação de tempo suficiente ao estudo e a atenção à análise de “casos negativos”, isto é, teorias que contradizem as descobertas (RUNESON; HöST, 2009).

3.9.2 Tipos de Serviços Sustentados pelos Times *DevOps*

Os times *DevOps* da instituição são responsáveis por sustentar diferentes tipos de serviços desenvolvidos e operados em arquiteturas de computação em nuvem. Esses serviços abrangem desde APIs de negócio até sistemas de processamento assíncrono, mecanismos de mensageria e fluxos de dados. A correta identificação dessas categorias é fundamental para determinar quais métricas DORA devem ser aplicadas em cada caso, levando em consideração aspectos como criticidade, padrões de operação e características específicas de execução. A Tabela 3.2 apresenta a classificação adotada neste trabalho.

Tipo de Serviço	Exemplo	Crítico	Métricas Aplicáveis
API REST	Serviço de autenticação	Sim	Frequência de implantações, Tempo de recuperação, Taxa de falhas
Processamento Batch	Fechamento financeiro diário	Sim	Tempo gasto para mudanças, Tempo de recuperação
Mensageria	Tópicos Kafka para eventos	Não	Frequência de implantações, Tempo de recuperação
Serviço de Dados	ETL em nuvem	Não	Tempo gasto para mudanças
Microserviço transacional	Pagamento instantâneo (PIX)	Sim	Todas as métricas DORA

Tabela 3.2: Classificação dos serviços sustentados pelos times *DevOps*.

Capítulo 4

Resultados

Nesta seção, apresentam-se os resultados obtidos em cada fase do trabalho, organizados conforme os objetivos específicos da pesquisa declarados no item 1.5.2. São descritos os métodos e técnicas empregados, as atividades realizadas e os resultados alcançados, seguindo a metodologia detalhada na tabela 3.1.

4.1 Estratégias de Pesquisa

A aplicação da abordagem metodológica adaptativa em atendimento ao OE1 permitiu contornar os desafios relacionados à escassez de publicações específicas sobre *DevOps* em grandes instituições financeiras. Os resultados da revisão da literatura foram organizados em duas estratégias complementares: *ad hoc* e *snowballing*, conforme descrito na seção 3.1. Foram estabelecidos critérios de inclusão e exclusão (Tabela 4.1) para garantir consistência e relevância dos artigos selecionados.

Critérios	Definição de Critério
CI01	Artigos que abordem princípios, métodos, práticas, técnicas ou estratégias de <i>DevOps</i> com ênfase em métricas DORA.
CI02	Artigos no idioma inglês.
CI03	Artigos publicados em conferências.
CE01	Artigos que passaram pelo processo de inclusão e exclusão ou que foram analisados.
CE02	Artigos que não contenham princípios, métodos, práticas, técnicas ou estratégias com ênfase em métricas DORA após leitura do resumo.
CE03	Artigos com conteúdos indisponíveis para visualização.
CE04	Artigos com o título e resumo que fogem da temática.
CE05	Artigos que não contenham princípios, métodos, práticas, técnicas ou estratégias de <i>DevOps</i> com ênfase em métricas DORA após leitura completa.
CE06	Artigos publicados antes de 2010.

Tabela 4.1: Lista dos Critérios de Inclusão e Exclusão

4.1.1 Evidências da Busca Inicial *Ad Hoc*

Durante a execução da pesquisa inicial, percebeu-se que o tema *DevOps* apresenta ampla aplicabilidade, sendo utilizado em empresas de diferentes portes e setores. A busca com palavras-chave como "Métricas DORA", "Performance DevOps", "Instituições Financeiras" e "Bancos" retornou poucos registros, predominando estudos voltados para *fintechs*.

Para ampliar o escopo, foram utilizadas combinações adicionais de termos, incluindo "DORA Metrics", "DORA Metrics AND DevOps", "DORA DevOps", "DevOps Maturity", "DevOps Performance" e "Financial DevOps". Essa segunda rodada de busca evidenciou limitações: muitos resultados referiam-se à *Declaration on Research Assessment (DORA)*, fora do escopo; outros estavam centrados em modelos de maturidade como CMMI; e alguns abordavam apenas otimização de recursos em nuvem ou ferramentas de segurança, sem relação direta com métricas DORA.

A pesquisa resultou em 73 artigos potencialmente relevantes, submetidos aos critérios de inclusão e exclusão descritos na Tabela 4.1. Após a filtragem, 14 artigos foram selecionados para análise detalhada, com concentração em pequenas e médias empresas e foco em aspectos culturais do *DevOps*.

Os temas mais recorrentes identificados incluíram:

- Métodos e benefícios do uso de métricas DORA;
- Barreiras culturais e organizacionais para a adoção do *DevOps*;
- Casos de estudo com ênfase em *startups* e *fintechs*.

4.1.2 Evidências da Busca Complementar por *Snowballing*

Diante da escassez de estudos diretamente relacionados ao setor financeiro, adotou-se a técnica de *snowballing* (WOHLIN, 2014), que permitiu ampliar o conjunto de artigos por meio da análise de referências complementares.

Essa abordagem possibilitou a identificação de 27 novos artigos, dos quais 18 foram incorporados após aplicação dos critérios de inclusão e exclusão. O processo foi conduzido em dois ciclos iterativos, garantindo consistência e rastreabilidade.

Entre os novos estudos identificados, destacaram-se:

- Estudos de caso sobre transformação *DevOps* em grandes empresas;
- Aplicações práticas de métricas DORA para avaliar desempenho organizacional em bancos;

- Estratégias de alinhamento entre práticas *DevOps* e regulamentos do setor financeiro.

Adicionalmente, foram identificadas lacunas na literatura relacionadas à mensuração quantitativa de desempenho *DevOps* em ambientes corporativos complexos, como instituições financeiras de grande porte. Para apoiar esse processo, utilizou-se a ferramenta *Connected Papers* (EITAN et al., 2024), além de mecanismos como *Google Scholar* e o Portal Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Portal Periódicos, 2020).

Ao final, foram analisados 32 artigos, que serviram como fundamentação teórica. A combinação entre revisão *ad hoc* e *snowballing* mostrou-se eficaz para superar as limitações impostas pela dispersão dos estudos sobre *DevOps* e métricas DORA no contexto financeiro.

Portanto, a decisão metodológica de estruturar a pesquisa em duas estratégias complementares garantiu consistência, ampliou o alcance da revisão e reforçou a validade do recorte adotado.

4.2 Algoritmos das Métricas

Para atender ao OE2, foram desenvolvidos algoritmos padronizados que suportam o cálculo das métricas DORA, assegurando consistência e confiabilidade nos resultados obtidos.

A definição desses algoritmos foi fundamentada em uma análise documental detalhada dos manuais técnicos das ferramentas disponíveis na instituição financeira em estudo, complementada por um levantamento das práticas amplamente adotadas na indústria, com base em publicações acadêmicas e especializadas. Durante essa análise, identificou-se uma forte correlação entre a abordagem proposta e o trabalho de Sanchez et al. (RUIZ et al., 2023), reforçando a validade dos métodos sugeridos para mensurar o desempenho dos times *DevOps*. Essa convergência evidencia que os algoritmos aqui apresentados estão alinhados com *benchmarks* reconhecidos no campo.

Para a formalização dos cálculos, empregou-se a lógica de primeira ordem, também conhecida como lógica de predicados, utilizada para quantificar atributos e expressar relações entre eles. Os resultados obtidos são apresentados a seguir.

Algoritmo para Frequência de implantações

1. Criar o conjunto I de implantações delimitado no período t_0 para t_n .
2. Calcular a frequência de implantações conforme:

$$\frac{|I|}{t_n - t_0} \quad (4.1)$$

Algoritmo para Tempo de Entrega para Mudanças

1. Criar o conjunto C de *commits* e o conjunto I de implantações.
2. Criar o conjunto P de todos os pares (c, i) , onde $c \in C$ e $i \in I$ tal que c está contido por i .
3. Criar o conjunto P_T de pares (t_i, t_c) para cada $(c, i) \in P$, onde t_i é o momento da implantação de i e t_c é o momento do *commit* de c .
4. Criar o conjunto T de tempos t para cada $(t_i, t_c) \in P_T$, onde $t = t_i - t_c$.
5. Calcular o prazo de entrega para mudanças conforme:

$$\frac{1}{|T|} \sum_{t \in T} t \quad (4.2)$$

Algoritmo para Tempo Médio para Recuperação

1. Criar o conjunto C de *commits* e o conjunto F de falhas.
2. Criar o conjunto P de todos os pares (c, f) , onde $c \in C$ e $f \in F$ tal que c corrige f .
3. Criar o conjunto I de todos os pares (i, f) , onde i é uma implantação que contém c e $(c, f) \in P$.
4. Criar o conjunto P_T de pares (t_i, t_f) para cada $(i, f) \in I$ onde t_i é o momento da implantação de i e t_f é o momento quando a falha é relatada de f .
5. Criar o conjunto T de tempos t para cada $(t_i, t_f) \in P_T$, onde $t = t_i - t_f$.
6. Calcular o tempo médio de recuperação conforme:

$$\frac{1}{|T|} \sum_{t \in T} t \quad (4.3)$$

Algoritmo para Taxa de Falhas

1. Criar o conjunto C de *commits*, o conjunto F de falhas e o conjunto I de implantações.
2. Criar o conjunto P de todos os pares (c, f) , onde $c \in C$ e $f \in F$ tal que c implica em f .
3. Criar o conjunto I_f de todos os pares (i, f) , onde i é uma implantação que contém c e $(c, f) \in P$.
4. Criar o conjunto I_e de todas as implantações exclusivas i , onde $(i, f) \in I_f$ e f é qualquer falha arbitrária.
5. Calcular taxa de falha de alteração conforme:

$$\frac{|I_e|}{|I|} \tag{4.4}$$

4.3 Funcionamento e Restrições do Modelo

Com o objetivo de atender ao OE3, foi realizado um levantamento detalhado dos pontos de coleta de dados necessários ao cálculo das métricas DORA. Nessa etapa, foram analisadas as possibilidades e limitações técnicas presentes no ambiente tecnológico da instituição financeira em estudo.

4.3.1 Ferramentas Candidatas

As soluções *DevOps* utilizadas na instituição financeira em estudo para automação, colaboração, implantação e melhoria contínua foram empregadas no fornecimento e tratamento dos dados. Para suprir os insumos necessários ao cálculo das métricas, além das informações provenientes das bases de dados, as seguintes ferramentas candidatas foram utilizadas para a extração:

1. *RTC*: utilizada para gerenciar as *sprints* em times ágeis;
2. *GitLab*: responsável pelo armazenamento e versionamento de código-fonte em repositórios *Git*²⁰, com recursos adicionais de acompanhamento de problemas e suporte a roteiros;

²⁰Sistema de controle de versão usado para rastrear mudanças em arquivos de computador. Seu objetivo é gerenciar quaisquer alterações feitas em um ou mais projetos em um determinado período de tempo. Auxilia na coordenação do trabalho entre os membros de um time de projeto e rastreia o progresso ao longo do tempo.

3. *Jenkins*: empregado na criação de *pipelines* e no gerenciamento de scripts de automação para compilações e publicações;
4. *Sonar*: utilizado para validação da cobertura de testes e verificação de boas práticas de codificação;
5. *JFrog*: destinado ao catálogo e gerenciamento das versões compiladas das soluções;
6. *GSTI*: gerenciador de serviços de TI, voltado principalmente à administração de mudanças nos ambientes e ao tratamento de incidentes.

As métricas de velocidade — frequência de implantações e tempo para mudanças — são extraídas exclusivamente do repositório de código-fonte, descrito em detalhes na seção seguinte. Ao passo que as métricas de estabilidade — tempo médio para recuperação e taxa de falhas — são obtidas a partir dos dados do *GSTI*, complementados por informações de falhas e *commits* de correção de *bugs* registrados no *GitLab*.

4.3.2 Extração das Métricas de Velocidade

A mensuração da frequência de implantações foi realizada a partir da leitura dos registros de histórias de usuário no *RTC*. A criação dessas histórias é responsabilidade dos *stakeholders*, que priorizam, dentro do backlog do time, aquelas que atendem às necessidades de negócio. A alteração do status da história de “nova” para “em andamento” marca o início da contabilização dessa métrica.

Durante o desenvolvimento, cada história é detalhada em tarefas, que organizam as entregas de código e vinculam os respectivos *commits*. Após o versionamento do código-fonte e a execução dos testes, a história é atualizada para “em homologação”. A atuação do P.O. nessa etapa altera o status para “homologada”, resultando em uma versão candidata à produção, denominada internamente como fase “intermediária”.

A partir desse ponto, conclui-se a etapa de desenvolvimento (Dev) e inicia-se a etapa de operações (Ops). Um artefato denominado item de entrega é criado e vinculado à história homologada, funcionando como um *ticket* de atendimento na esteira. Nas operações, a versão candidata é compilada via *Jenkins* e armazenada no catálogo por meio do *JFrog*. O script de compilação no *Jenkins* inclui etapas de validação da qualidade do código, acionando a ferramenta *Sonar*. O *Sonar* gera relatórios com a pontuação de qualidade da solução e o percentual de cobertura de testes.

Essas validações de qualidade, que determinam se uma versão pode ou não ser publicada em produção, estão sintetizadas na Tabela 4.2. Ela apresenta os critérios utilizados, suas variações possíveis e os valores mínimos ou obrigatórios exigidos para aprovação.

Validações	Variação	Valor mínimo ou obrigatório
Cobertura de código novo	0 a 100%	45%
Teste de interface ou API	Passou/Falhou	Passou
Cobertura de código total	0 a 100%	60%
Categoria de ocorrências no código	0 - infos 1 - menores 2 - maiores 3 - blocantes 4 - críticos	\leq maiores
Vulnerabilidades de segurança	A, B, C, D e E	A
Análise de dependências	baixo, médio, alto, crítico	médio

Tabela 4.2: Critérios de validação de qualidade do *software*

A integração desses critérios ao processo de entrega garante que apenas versões que atendam aos padrões mínimos de qualidade avancem para produção. Dessa forma, a métrica de velocidade não se limita à frequência de implantações, mas reflete também a capacidade do time em manter níveis consistentes de qualidade durante o ciclo de desenvolvimento e operações.

Caso o item de entrega seja aprovado nas validações automatizadas da esteira, o envio da solução para produção é acordado entre o Gestor e a área de tecnologia responsável. A liberação é realizada por um profissional distinto daquele que registrou o *ticket* de atendimento. Dependendo da nota atribuída pelo motor de liberação, exige-se um nível hierárquico específico da pessoa responsável, em relação inversa: quanto maior a nota, menor o nível hierárquico requerido.

Para não comprometer o indicador da métrica de frequência de implantações, adota-se a prática de realizar o envio o mais tempestivamente possível. Apenas em casos de incidentes as versões candidatas permanecem em espera antes da publicação. Uma vez publicada a nova versão, independentemente do uso da estratégia *blue/green*, a história do usuário é atualizada para o status “em produção”. Nesse momento, juntamente com a atualização do registro, conclui-se a mensuração dessa métrica.

No que se refere à métrica de tempo gasto para mudança (*lead time*), conforme definido pelas métricas DORA, sua medição inicia-se no momento em que o desenvolvedor começa a trabalhar em uma alteração e se encerra quando ela é enviada para produção. Esse intervalo pode ser decomposto em duas etapas: o tempo dedicado à implementação da mudança e o tempo consumido pelo processo de implantação até a produção. A análise evidencia que a fase de implantação tende a ser mais longa, o que justifica esforços contínuos de otimização.

O tempo gasto para mudança é considerado uma métrica crítica, pois revela a capacidade de um time em responder rapidamente a novas condições, eventos ou necessidades.

Por exemplo, diante da detecção de um defeito, avalia-se a rapidez com que o time consegue criar uma correção e disponibilizá-la em produção. Da mesma forma, quando há demanda por uma nova funcionalidade ou melhoria incremental, a agilidade na entrega torna-se fator decisivo. Organizações capazes de disponibilizar mudanças em ciclos curtos tendem a ser mais bem-sucedidas do que aquelas que levam meses para colocar qualquer alteração em produção (FORSGREN; HUMBLE; KIM, 2018).

No contexto da instituição financeira em estudo, as métricas de velocidade passaram a ser mensuradas por meio da interação entre as ferramentas descritas. A métrica de frequência de implantações contabiliza o número de entregas realizadas, enquanto a métrica de tempo gasto para mudança mensura a duração do ciclo. Conforme mencionado anteriormente, a fase denominada “intermediária” também é acompanhada como parte do cálculo parcial do tempo gasto para mudança.

4.3.3 Extração das Métricas de Estabilidade

Os microsserviços desenvolvidos na instituição objeto desta pesquisa são configurados com mecanismos de alerta que, em caso de indisponibilidade, geram automaticamente registros de incidentes. A ferramenta *GSTI*, conforme mencionado anteriormente, é responsável pela administração desses registros. As soluções são parametrizadas para disparar incidentes contendo informações detalhadas sobre o microsserviço afetado, a funcionalidade impactada e o log correspondente ao momento da falha.

Uma vez gerado o registro de incidente, a gerência responsável pela solução é notificada e direciona o time *DevOps* pertinente para atuação. Além da quantidade de incidentes registrados em determinado período pelo *GSTI*, considera-se também o número de entregas em produção informado pela ferramenta *GitLab*, de modo a consolidar a métrica de taxa de falhas.

A quarta métrica, referente ao tempo médio de recuperação, utiliza como insumo o instante de abertura do registro de incidente. A partir desse momento, mesmo sem atendimento imediato, o tempo passa a ser contabilizado. Essa métrica integra o processo de resposta a incidentes e avalia a rapidez com que uma mudança corretiva é implementada e disponibilizada em produção após a identificação da falha.

4.3.4 Avaliação do Modelo com Especialistas

Por meio da técnica de entrevista aberta, foram entrevistados quatro especialistas: dois da gerência de construção, um da gerência de infraestrutura e outro da gerência de monitoração. Essas entrevistas exploraram a estrutura e o comportamento dos times *DevOps*, além dos dados relevantes para as métricas DORA. A compreensão da cultura dos times

foi essencial para assegurar a escolha apropriada dos dados, o que resultou na revisão de atributos selecionados e no descarte de elementos considerados desnecessários.

Os especialistas compartilharam desafios e expectativas em relação às métricas DORA. Um dos desafios mais apontados refere-se à necessidade de adaptação das métricas à realidade dos times. Como observado por um especialista em infraestrutura:

O uso das métricas DORA pode permitir que possamos identificar não só os problemas atuais, mas também prever situações de falha com base nos padrões históricos de dados. Isso aumentaria nossa capacidade de resposta e a eficiência na resolução de incidentes.

(Especialista de infraestrutura, entrevista pessoal, 2024).

Além disso, outro especialista destacou a importância de consolidar as métricas como parte do processo contínuo de evolução dos times:

Precisamos consolidar as métricas como parte integrante do nosso processo de evolução contínua. Elas nos ajudam a entender onde estão os gargalos e onde podemos melhorar. A chave é integrar essas métricas ao nosso trabalho diário.

(Especialista A da gerência de construção, entrevista pessoal, 2024).

Por sua vez, o especialista de monitoração enfatizou que a aplicação das métricas DORA pode transformar a abordagem atual, ao permitir que times avancem de um modelo reativo para um modelo preditivo. Ele observou:

As métricas DORA representam uma oportunidade de evolução para o monitoramento. Elas não apenas ajudam a identificar problemas em tempo real, mas possibilitam a antecipação de falhas com base em padrões de comportamento e dados históricos. Isso significa que poderemos atuar preventivamente, reduzindo o impacto de incidentes e aumentando a confiabilidade dos serviços entregues. Essa transição para uma abordagem preditiva tem o potencial de redefinir os padrões de qualidade e eficiência nos processos de monitoramento.

(Especialista de monitoração, entrevista pessoal, 2024).

Além disso, o especialista B da gerência de construção destacou como as métricas podem ser transformadoras quando utilizadas como um alicerce para decisões estratégicas:

As métricas DORA nos fornecem um panorama claro sobre o desempenho e a capacidade do time de entregar valor de maneira consistente. Porém, o verdadeiro valor está em como interpretamos esses números para ajustar nossa abordagem. Por exemplo, ao identificar gargalos em um fluxo de trabalho, podemos priorizar melhorias que aumentem a eficiência e reduzam a sobrecarga do time. É essa capacidade de direcionar esforços com base em dados reais que torna as métricas tão valiosas.

(Especialista B da gerência de construção, entrevista pessoal, 2024).

²⁰Entrevista realizada com o especialista de infraestrutura em Brasília, no contexto da pesquisa sobre a aplicação das métricas DORA para aprimoramento de práticas *DevOps*, em 2024.

Sobre Times *DevOps*

Os especialistas da gerência de construção descreveram a composição típica dos times, composta por um proprietário de produto, um agente de aceleração, um a dois designers de UX e seis a sete engenheiros de *software*, além de um profissional especializado na coleta e exposição de dados dos aplicativos. A metodologia de trabalho pode ser resumida nos seguintes pontos:

- A comunicação é facilitada por meio de reuniões diárias no *Microsoft Teams*.
- As atividades são organizadas em *sprints* de duas semanas, com reuniões relacionadas à *sprint* corrente a cada duas semanas.
- Realizam-se pontos de controle semanais para avaliar o progresso do time.
- O desenvolvimento é gerenciado por meio de um *backlog*, com lançamento de recursos (concluídos e parcialmente concluídos) e correções de *bugs* semanalmente, visando entregar a maioria dos *commits* dentro de uma semana.
- Os recursos não prontos para lançamento ficam ocultos por sinalizadores.
- As alterações são implantadas em um ambiente piloto para testes, semelhante ao ambiente de produção, antes do lançamento.
- Semanalmente, é realizada uma sessão de testes de execuções encadeadas que contemplam todas as funcionalidades (testes de regressão).
- Os lançamentos são realizados de maneira programada, com congelamentos pré-definidos, e adiamentos em caso de problemas durante a sessão de testes.
- Quanto ao processo de liberação, procuram manter o ramo principal disponível para entregas oportunas.

Sobre os times da gerência de infraestrutura, o especialista compartilhou detalhes sobre a composição e práticas desses times, que incluem aproximadamente quinze engenheiros de *software*, de um a três especialistas em operações, um engenheiro de controle de qualidade e um proprietário de produto. Suas práticas de trabalho podem ser resumidas da seguinte forma:

- Adotam um processo de *sprint* flexível, semelhante ao *Kanban*.
- Seguem rotinas *Scrum*, embora de forma menos estruturada.
- Abordam o trabalho de maneira investigativa, frequentemente abrindo tarefas de análise de cenários, fluxos ou situações que precisem de correção ou evolução.

- Com a ajuda do engenheiro de controle de qualidade, realizam testes de integração e de lançamento.
- Não seguem uma data fixa para lançamento, optando por fazê-lo quando oportuno.
- Buscam implantar em produção com a maior frequência possível.
- Agrupam recursos e correções de *bugs* em versões.
- Facilitam pontos de controle entre times, promovendo a discussão e o compartilhamento de estudos e práticas relacionadas à configuração dos produtos.

Sobre a Seleção e Qualidade dos Dados para Métricas DORA

Todos compartilharam algumas ideias gerais para subsidiar as métricas DORA: necessidade de uma abordagem de entrega contínua para que os resultados sejam autênticos, consideração de fatores externos ao controle do time, como dependências de outras áreas, débitos técnicos, atuações extraordinárias, sobrecarga do time, assim como períodos de ausências dos integrantes.

Dados do registro aprovado de publicação foram indicados para mensurar a quantidade de implantações, considerando o momento que fora executado. A quantidade de *commits* será resgatada pelos repositórios do código-fonte. Para referência de entrega, o momento de início de atendimento das histórias de usuários existentes na *sprint* do time será considerado. Para falhas, os especialistas entendem que o registro de incidente fornece todos os dados necessários, tanto para tempo médio de recuperação, quanto a taxa de falhas.

Durante a discussão sobre o processo de lançamento, com foco na qualidade da seleção dos dados, foi observada a retenção frequente de uma versão não lançada enquanto aguardavam homologações em cenários específicos. Essa situação afeta os objetivos de implantação em produção e a meta de tempo de mudança, que é de pelo menos uma semana.

Sobre a Expectativa de Uso do Modelo

As entrevistas revelaram uma variedade de abordagens e desafios enfrentados pelos times em relação ao acompanhamento de desempenho *DevOps*, incluindo a utilização de indicadores de desempenho contínuo, a implementação de ferramentas de automação para integração e entrega contínua (CI/CD), práticas de monitoramento de qualidade e segurança, além de iniciativas de capacitação e alinhamento cultural para adoção de práticas *DevOps*.

Ainda há espaço para melhorias e aprendizado contínuo neste campo. A aplicação das métricas DORA exige não apenas o domínio técnico para coleta e análise de

dados, mas também uma mudança cultural significativa dentro dos times. Precisamos fomentar um ambiente onde os times compreendam o valor dessas métricas, não apenas como indicadores de desempenho, mas como ferramentas para guiar decisões estratégicas e operacionais. Além disso, é essencial criar processos que integrem essas práticas ao fluxo de trabalho diário, de forma que se tornem uma parte natural da dinâmica dos times, promovendo ciclos de melhoria contínua.

(Especialista de monitoração, entrevista pessoal, 2024).

Para utilização plena do modelo, é recomendado que os times adote um modelo de governança de dados robusto, com acompanhamento constante e feedback contínuo, ajustando as métricas conforme as necessidades específicas de cada time.

4.3.5 Validação Qualitativa e Triangulação dos Dados

Com o objetivo de assegurar maior robustez e confiabilidade aos resultados obtidos, foi realizada uma triangulação qualitativa entre os dados quantitativos das métricas DORA, as entrevistas conduzidas com especialistas e as percepções práticas relatadas pelos times envolvidos. Essa abordagem permitiu integrar diferentes fontes de evidência, reduzindo vieses e ampliando a compreensão sobre a maturidade dos processos *DevOps* na organização analisada.

A integração entre métricas e entrevistas revelou que os indicadores de *frequência de implantações* e *tempo de entrega para mudanças* refletem diretamente os desafios apontados pelos especialistas da gerência de construção e infraestrutura, como gargalos em homologação e dependências externas. Essa convergência confirma a aderência dos valores calculados à realidade operacional.

De forma complementar, os resultados de *taxa de falhas* e *tempo médio de recuperação* corroboraram a percepção dos especialistas sobre a necessidade de práticas preventivas e maior integração entre monitoramento e desenvolvimento. A triangulação demonstrou que os indicadores não apenas mensuram desempenho técnico, mas também refletem aspectos culturais e organizacionais discutidos nas entrevistas, reforçando a importância da consolidação das métricas como parte de um processo contínuo de evolução.

A síntese dos achados pode ser observada na Tabela 4.3, que apresenta a relação entre cada métrica DORA, os principais pontos levantados nas entrevistas e as percepções dos especialistas. Essa integração evidencia que a maturidade *DevOps* deve ser compreendida como um fenômeno multidimensional, envolvendo tanto indicadores técnicos quanto fatores humanos e culturais. A triangulação qualitativa, portanto, fortalece a confiabilidade das conclusões e oferece uma visão integrada sobre os avanços e desafios na adoção das práticas *DevOps*.

²⁰Entrevistas realizadas com especialistas das gerências de infraestrutura, construção e monitoração em Brasília, no contexto da pesquisa sobre a aplicação das métricas DORA, em 2024.

Métricas DORA	Entrevistas	Percepções dos Especialistas
frequência de implantações	Gargalos em homologação	Indicadores refletem dependências externas
Tempo de entrega para mudanças	Necessidade de maior agilidade	Confirma desafios de coordenação entre áreas
Taxa de falhas	Preocupação com incidentes	Ênfase em práticas preventivas e monitoramento
Tempo médio de recuperação	Capacidade de resposta	Potencial de evolução para abordagem preditiva

Tabela 4.3: Integração qualitativa entre métricas DORA, entrevistas e percepções dos especialistas.

4.3.6 Restrições do Modelo

A análise inicial do cenário existente sobre o uso das métricas DORA permitiu identificar restrições relevantes para a mobilização do processo de extração:

- **Informações dispersas:** em organizações de grande porte, os dados encontram-se distribuídos em múltiplas fontes, o que dificulta o levantamento completo das informações necessárias ao cálculo das métricas.
- **Extração e processamento de dados:** é comum que os dados disponíveis não estejam refinados para fins de monitoramento, exigindo etapas adicionais de preparação e tratamento antes da análise.
- **Interpretação das informações:** o contexto é essencial para a correta leitura das métricas, sendo necessário considerar a realidade dos produtos sustentados por cada time. A partir dessa análise, define-se a forma adequada de interpretar os resultados.

As restrições identificadas foram mitigadas por meio da padronização do uso das ferramentas e da uniformização dos dados coletados e analisados. Contudo, devido à limitação da amostra, não foi possível validar, em nível corporativo, se o modelo atende a todos os tipos de times *DevOps*. Assim, compreende-se que cada time deverá interpretar os resultados conforme seu contexto específico. Com a aplicação em maior escala, espera-se que novas versões do serviço sejam desenvolvidas para contemplar um espectro mais amplo de necessidades.

No que se refere às métricas de estabilidade, não foi possível automatizar o processo de extração. As ferramentas utilizadas para o gerenciamento de incidentes são proprietárias, e a implementação de APIs que sustentem o serviço dependeria de negociação com os

fornecedores. Diante dessa limitação, as métricas de “tempo de recuperação” e “taxa de falhas” foram coletadas manualmente para este estudo.

4.3.7 Considerações sobre APIs e Extração de Logs

A Tabela 4.4 sintetiza os principais critérios de comparação entre APIs e logs, como forma de coleta, granularidade, dependência tecnológica, flexibilidade, complexidade de integração, custos operacionais, segurança, retenção e adequação às métricas DORA.

Critério	APIs como Fonte das Métricas	Logs como Fonte das Métricas
Forma de Coleta	Dados agregados e expostos por endpoints definidos pelo fornecedor.	Registros brutos gerados por serviços, pipelines e infraestrutura.
Granularidade	Limitada ao nível de detalhe disponibilizado pelo endpoint.	Alta granularidade; permite reconstrução completa de eventos.
Dependência de Fornecedor	Alta dependência, pois os dados seguem restrições de versão e políticas do provedor.	Menor dependência; logs podem ser armazenados e processados internamente.
Flexibilidade para Produção de Métricas	Baixa flexibilidade; métricas derivadas dependem da oferta do provedor.	Alta flexibilidade; permite criação de métricas específicas e filtragens detalhadas.
Complexidade de Integração	Integração simplificada, com padrão estável e documentação fixa.	Exige coleta, normalização, limpeza e padronização dos registros.
Custos Operacionais	Baixo custo operacional de processamento, pois os dados chegam prontos.	Maior custo de armazenamento, processamento e governança.
Segurança e Acesso	Exposição controlada pelo provedor; menor risco de vazamento de informações internas.	Pode conter dados sensíveis (configuração, caminhos internos, mensagens de erro).
Retenção	Limitada às políticas do provedor; nem sempre atende necessidades históricas.	Configurável pela organização, permitindo retenção estendida ou completa.
Adequação para Métricas DORA	Adequada quando a necessidade é um conjunto fixo de indicadores.	Adequada quando há necessidade de auditoria detalhada, reconstrução temporal ou métricas adicionais.

Tabela 4.4: Comparação entre uso de APIs e extração de logs como fonte para mensuração das métricas DORA.

Neste trabalho, a recuperação das métricas DORA é realizada exclusivamente por meio de APIs disponibilizadas pelo provedor da plataforma de entrega. Essa abordagem oferece dados estruturados e normalizados, reduzindo o esforço de processamento e simplificando a integração com os sistemas internos de análise. No entanto, sua eficácia está condicionada à disponibilidade, estabilidade e políticas de retenção definidas pelo fornecedor, limitando a granularidade das medições ao nível exposto pelos endpoints.

Como alternativa, a extração direta de logs permite acesso a registros detalhados de eventos de implantação, falhas, tempos de execução e operações internas do pipeline. Essa granularidade viabiliza a reconstrução de séries temporais completas e a produção de métricas derivadas não oferecidas nativamente pelas APIs. Além disso, reduz a dependência tecnológica do fornecedor, pois os registros podem ser armazenados e processados em infraestrutura interna. Por outro lado, exige mecanismos adicionais de coleta, normalização e governança, além de controles de segurança mais rigorosos, devido ao potencial de exposição de informações sensíveis.

Dessa forma, a escolha entre APIs e logs deve considerar fatores como simplicidade de integração, nível de granularidade desejado, independência tecnológica e políticas de segurança da organização. Ambas as abordagens são viáveis para mensuração das métricas DORA, mas apresentam implicações distintas quanto ao custo operacional, precisão das medições e flexibilidade para evolução do modelo analítico.

4.4 Levantamento de Requisitos

Com o objetivo de atender ao OE4, foi realizada a atividade de levantamento dos requisitos funcionais e não funcionais do artefato. Essa etapa antecede a modelagem arquitetural da solução e tem como finalidade explicitar as necessidades que o sistema deve atender, garantindo que os diagramas e modelos propostos estejam fundamentados em requisitos claros e verificáveis. Para os requisitos não funcionais, adotou-se a metodologia URPS+ (CYSNEIROS, 2001; BASS; CLEMENTS; KAZMAN, 2012), que organiza atributos de qualidade em categorias sistemáticas, assegurando abrangência e consistência na análise.

4.4.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades necessárias para suportar o cálculo das métricas DORA e a avaliação do desempenho *DevOps*. Entre os principais requisitos identificados estão:

- RF1 – Capturar dados de commits e implantações a partir dos repositórios de código-fonte.
- RF2 – Calcular a frequência de implantações em determinado período.
- RF3 – Calcular o tempo de entrega para mudanças (*lead time*).
- RF4 – Calcular o tempo médio de recuperação de falhas.
- RF5 – Calcular a taxa de falhas por implantação.

- RF6 – Consolidar os resultados em relatórios acessíveis aos times *DevOps*.
- RF7 – Integrar-se às ferramentas utilizadas pela instituição (RTC, GitLab, Jenkins, Sonar, JFrog, GSTI).

4.4.2 Requisitos Não Funcionais

A classificação dos requisitos não funcionais foi realizada com base na metodologia URPS+, que organiza atributos de qualidade em categorias específicas. A Tabela 4.5 sintetiza os resultados dessa análise, evidenciando aspectos de usabilidade, confiabilidade, desempenho, suportabilidade e atributos adicionais.

Categoria URPS+	Requisitos Identificados
Usabilidade (U)	Interfaces claras e acessíveis para consulta das métricas.
Confiabilidade (R)	Disponibilidade contínua do serviço; tolerância a falhas na coleta de dados.
Performance (P)	Processamento das métricas em tempo hábil; resposta rápida às consultas.
Suportabilidade (S)	Escalabilidade para múltiplos times; facilidade de manutenção e evolução.
Atributos adicionais (+)	Segurança dos dados; conformidade regulatória; interoperabilidade com ferramentas de CI/CD.

Tabela 4.5: Classificação dos requisitos não funcionais segundo URPS+

O resultado consolidado desta atividade é o Catálogo de Requisitos (R4), composto pelos requisitos funcionais (RF1–RF7) e pelos requisitos não funcionais classificados segundo a metodologia URPS+. Esse catálogo servirá como insumo direto para a criação do modelo arquitetural da solução (A7), estabelecendo a base conceitual que orienta a modelagem e garantindo que os diagramas propostos estejam alinhados às funcionalidades esperadas e às restrições de qualidade identificadas. Dessa forma, a arquitetura resultante reflete não apenas os cálculos das métricas DORA, mas também atributos de usabilidade, confiabilidade, desempenho, suportabilidade e requisitos adicionais.

4.5 Arquitetura da Solução

Na sequência, em cumprimento ao OE4, foi proposta uma versão preliminar do modelo das APIs, fundamentada no padrão arquitetural MVC, conforme descrito na Seção 3.8.1 da metodologia. Nesse modelo, a camada de recurso, ou *resource (view)*, desempenha o papel de porta de entrada para as requisições realizadas pelos usuários ou sistemas externos. Esta camada é responsável por interpretar as solicitações e direcioná-las de maneira adequada para a camada de serviço, ou *service (controller)*.

A aplicação prática do padrão MVC demonstrou benefícios concretos na prova de conceito. A separação clara entre camadas facilitou a implementação incremental das APIs, permitindo que ajustes fossem realizados em cada componente sem comprometer o funcionamento global. Essa modularidade reduziu a complexidade de manutenção durante os testes e favoreceu a integração com sistemas existentes na instituição. Além disso, a familiaridade da equipe com o uso do MVC acelerou o desenvolvimento e diminuiu a curva de aprendizado, confirmando que a decisão metodológica descrita anteriormente resultou em ganhos efetivos de eficiência e confiabilidade no ambiente experimental.

A camada de serviço centraliza a lógica de negócio, validando as entradas e coordenando as operações necessárias para atender às requisições. Além disso, ela atua como intermediária na comunicação com a camada de dados, ou *model*. Essa última camada é responsável pela interação direta com os repositórios de dados, abstraindo os detalhes de acesso e persistência.

Propõe-se o uso do estilo arquitetural REST²¹, que define restrições e princípios aplicáveis na criação de serviços da web. O REST utiliza métodos padrões HTTP, como GET, POST e DELETE, o que simplifica sua utilização e garante interoperabilidade entre sistemas.

Nesse contexto, o padrão MVC e o estilo REST não concorrem, mas se complementam. O MVC organiza a aplicação internamente em camadas de responsabilidade, enquanto o REST estabelece como essas funcionalidades são expostas externamente por meio das APIs. Assim, o *Controller* recebe requisições REST, o *Model* processa os dados e a *View* retorna a resposta serializada (como JSON ou XML). Essa integração entre MVC e REST assegura tanto a clareza estrutural da aplicação quanto a padronização da comunicação entre sistemas distribuídos.

Para definir o funcionamento das APIs, foram elaborados três diagramas de classe:

1. A visão do *resource*, que apresenta como as funcionalidades são oferecidas nas chamadas das APIs e os retornos esperados.
2. A visão do repositório, que evidencia o consumo dos *commits*, *deploys* e *tags* das versões do código-fonte pela camada de serviço.
3. A visão do cluster, que mostra o consumo dos *namespaces*, *containers* e *pods* dos microsserviços disponíveis.

²¹O REST (Representational State Transfer) é um estilo arquitetônico aplicado para fornecer padrões entre sistemas de computador na web, facilitando a comunicação entre eles. A implementação do cliente e do servidor pode ser feita de forma independente, sem que cada um conheça o outro.

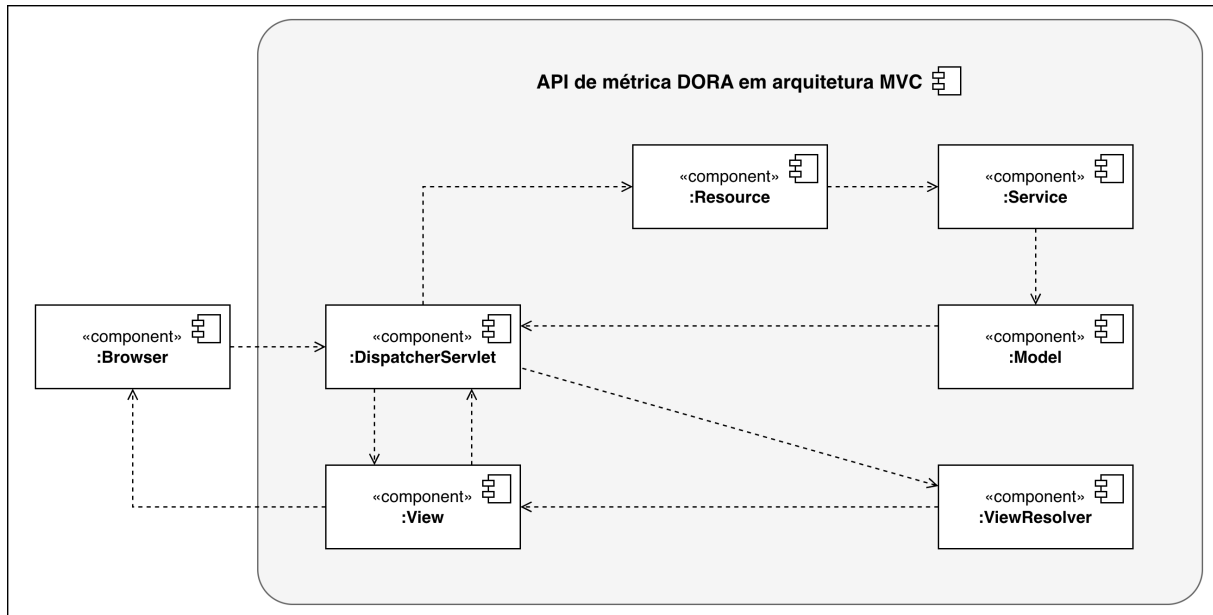


Figura 4.1: Diagrama de Componentes das Camadas de API e do Consumidor.

O diagrama UML *Visão resource* resume a arquitetura de serviços para análise de desempenho de entregas de software. A seguir, uma visão simplificada dos principais elementos.

- **Recursos (Resources):**
 - **FrequenciaEntregaResource** – expõe API para frequência de deploys.
 - **TempoEntregaResource** – expõe API para tempo de entrega.
- **Serviços (Services):**
 - **FrequenciaEntregaService** – cálculo da frequência de deploys.
 - **TempoEntregaService** – estatísticas de tempo de entrega e manipulação de dados.
- **Classificações (Enums):**
 - **FrequenciaEntregaPerformanceRating** – avalia periodicidade (diário, semanal, mensal, etc.).
 - **TempoEntregaPerformanceRating** – avalia tempo de entrega (diário, semanal, mensal, etc.).
- **Infraestrutura:**
 - **AppService** – inicializa contexto da aplicação.

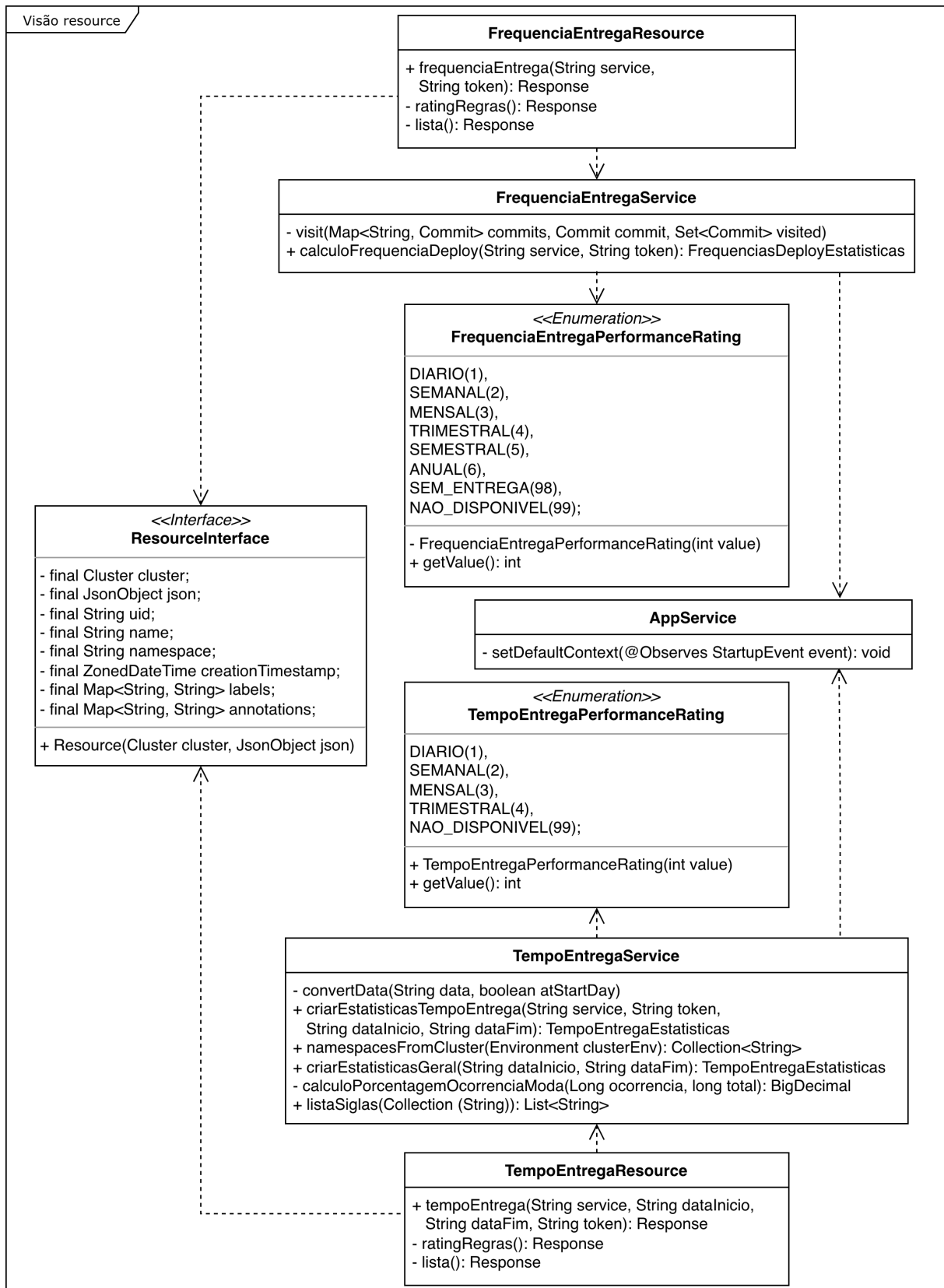


Figura 4.2: Diagrama de Classes da Visão *Resource*.

- **ResourceInterface** – abstração comum para recursos de cluster.

- **Relações:**

- Recursos invocam serviços para lógica de negócio.

- Serviços utilizam enums para classificar resultados.

- **AppService** configura ambiente de execução.

- **ResourceInterface** padroniza manipulação de recursos.

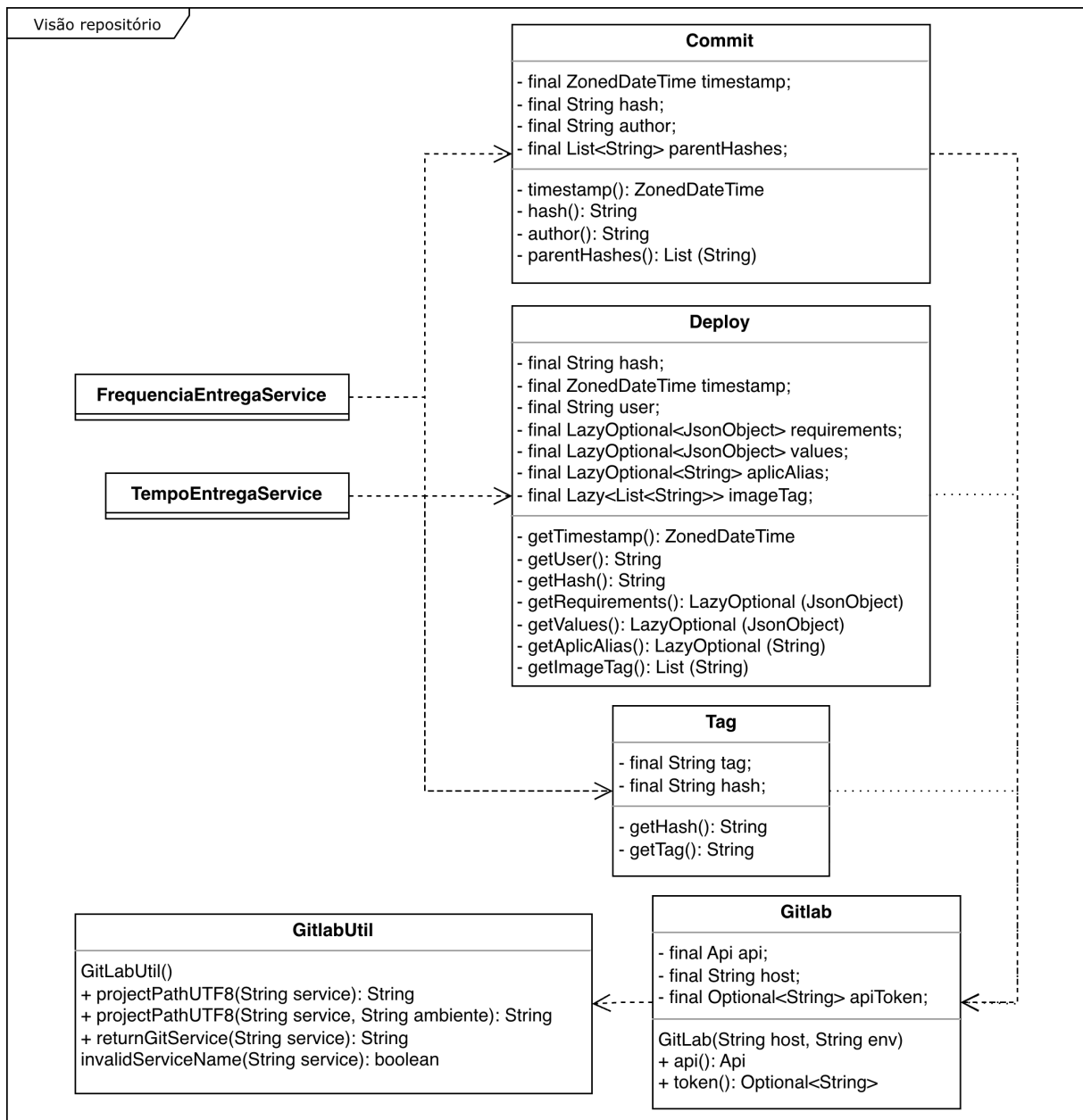


Figura 4.3: Diagrama de Classes da Visão Repositório.

O diagrama UML da *Visão Repositório* apresenta os elementos centrais para análise de commits, deploys e tags em sistemas de versionamento e entrega contínua.

- **Entidades de Repositório:**

- **Commit** – representa uma alteração no código com autor, hash, timestamp e referências aos commits anteriores.
- **Deploy** – representa uma entrega com informações de usuário, timestamp, requisitos, valores e imagens associadas.
- **Tag** – vincula um identificador textual a um commit específico.

- **Utilitários GitLab:**

- **GitlabUtil** – fornece métodos auxiliares para manipulação de caminhos e validação de serviços.
- **Gitlab** – encapsula acesso à API do GitLab, incluindo host e token de autenticação.

- **Serviços de Métricas:**

- **FrequenciaEntregaService** – utiliza dados de deploy para calcular frequência de entregas.
- **TempoEntregaService** – analisa tempo entre commits e deploys para gerar estatísticas.

- **Relações:**

- Deploy e Tag referenciam Commit por meio do atributo hash.
- Serviços de métrica consomem dados de Deploy para gerar indicadores.
- GitlabUtil e Gitlab oferecem suporte à integração com repositórios externos.

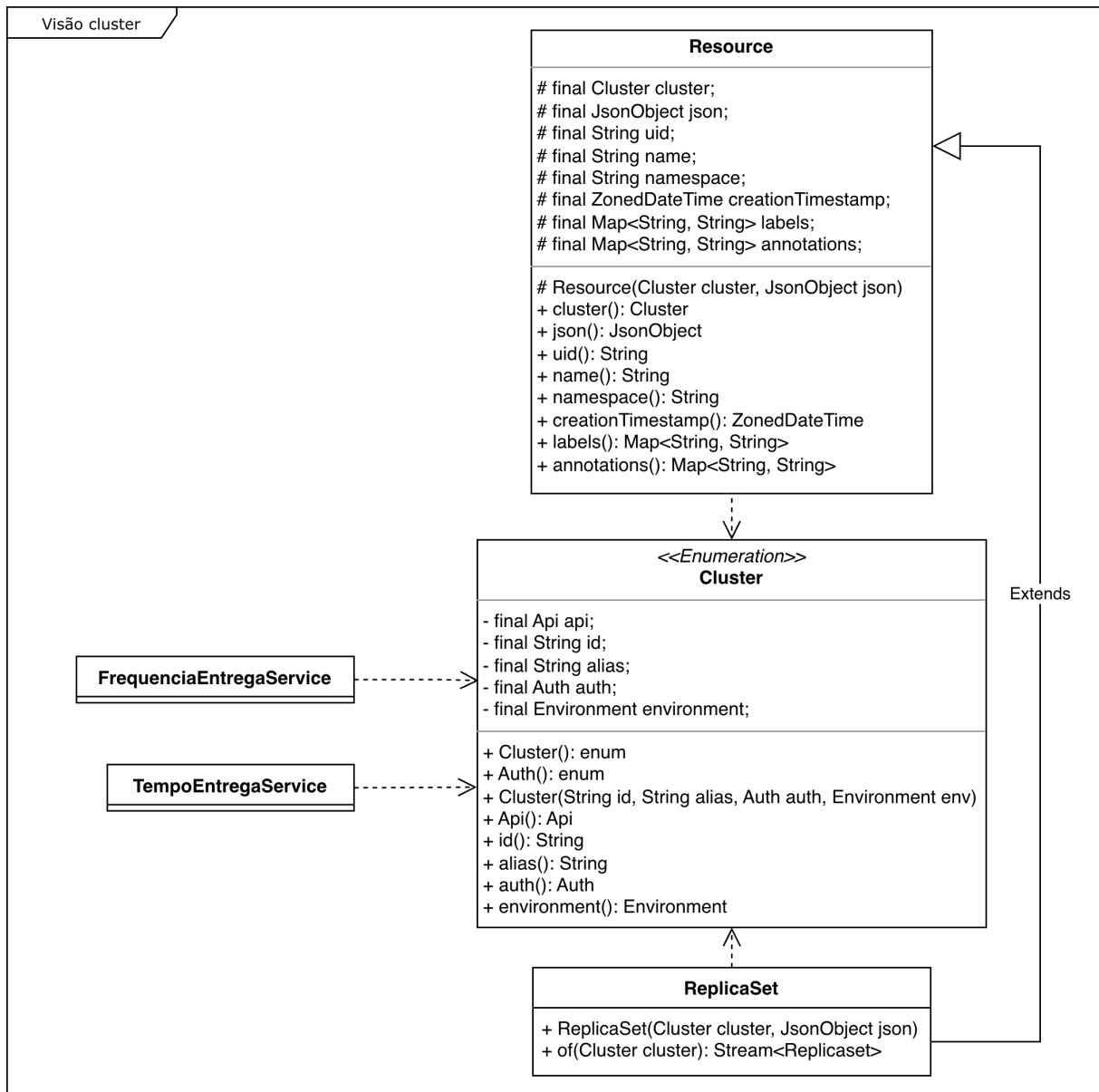


Figura 4.4: Diagrama de Classes da Visão *Cluster*.

O diagrama UML da *Visão Cluster* apresenta os elementos centrais para representação de recursos e serviços em ambientes de cluster.

- **Recursos de Cluster:**

- **Resource** – representa um recurso genérico com metadados como UID, nome, namespace, labels e timestamp.
- **ReplicaSet** – estrutura replicável associada a um cluster, com métodos de construção e filtragem.

- **Configuração de Cluster:**

- **Cluster** – enumeração que define atributos como `id`, `alias`, `auth` e `environment`, além de acesso à API.

- **Serviços de Métricas:**

- **FrequenciaEntregaService** – utiliza dados de `Cluster` para calcular frequência de entregas.
- **TempoEntregaService** – consome dados de `Cluster` para gerar estatísticas de tempo de entrega.

- **Relações:**

- `Resource` estende `ReplicaSet` e utiliza `Cluster` como referência.
- Serviços de métrica dependem de `Cluster` para acessar dados e gerar indicadores.

No *cluster*, foi possível verificar se a entrega de uma versão foi efetivamente disponibilizada, além de analisar as características, o consumo e o comportamento dos micro-serviços.

4.6 Código-fonte das APIs

O código-fonte das classes essenciais para o funcionamento das APIs responsáveis pelo cálculo das métricas DORA de tempo de entrega e frequência de implantações, em conformidade com o OE5, encontra-se disponível publicamente no repositório GitHub:

<<https://github.com/EmanuelDF/devops-metrics-dora>>

4.7 Avaliação do Desempenho *DevOps*

Esta etapa avalia o desempenho dos times, identificando oportunidades de melhoria para entregas mais ágeis e seguras, alinhadas aos objetivos organizacionais. Foram analisadas as métricas DORA e aplicado o *DORA DevOps Quick Check*, em atendimento ao OE6, com o propósito de verificar a eficácia do modelo proposto. Os resultados estabelecem uma linha de base para futuras comparações e orientam ações voltadas à evolução do desempenho dos times *DevOps*.

4.7.1 Componentes do Estudo

Os insumos definidos como componentes de estudo passaram por constante refinamento ao longo da pesquisa. A análise dos resultados influenciou o versionamento desses componentes, buscando maior qualidade no estudo de caso. O objetivo é fornecer evidências sobre o desempenho de times *DevOps* em instituições financeiras e propor uma forma sistemática de mensuração.

Questões de Estudo

Espera-se que o uso das métricas DORA contribua para ganhos de desempenho, maior qualidade e tempestividade na entrega de produtos. As proposições de estudo incluem:

- Compreender o desempenho de um time *DevOps*;
- Apoiar decisões de melhoria baseadas em métricas;
- Identificar oportunidades de evolução nos processos;
- Ampliar o suporte às soluções;
- Reduzir o tempo de entrega e aumentar a regularidade das implantações;
- Melhorar a percepção dos clientes sobre os serviços;
- Motivar os times *DevOps* em suas atividades.

Com base nesse escopo, foram definidas as seguintes questões de estudo:

Questões	Definição de Questão
Q01	Como recuperar a métrica de frequência de implantações?
Q02	Como coletar o número de implementações em um intervalo de tempo especificado?
Q03	Como recuperar a métrica de tempo gasto para mudanças?
Q04	Como recuperar a métrica de tempo de recuperação?
Q05	Como recuperar a métrica de taxa de falhas?
Q06	Quais foram os ganhos que as métricas DORA trouxeram na prática para o time?
Q07	Qual a percepção dos gestores de produto sobre a qualidade e volume de entrega das soluções nos times <i>DevOps</i> que usam as métricas DORA?

Tabela 4.6: Lista das Questões de Estudo

Será investigada a forma como o mercado contabiliza as métricas DORA, a fim de orientar o processo de extração na organização analisada. Seguindo o padrão observado

em artigos acadêmicos e técnicos, a unidade de análise é o time *DevOps*. A mensuração das métricas pode ser realizada em diferentes granularidades: meses, semanas, dias ou horas.

Com base na Tabela 2.2, será avaliado o nível de desempenho do time *DevOps* objeto da amostra. Serão identificados os fatores que conduzem ao nível resultante. Na validação, será considerada a percepção prática do time em relação a cada métrica. Caso sejam mobilizadas ações de melhoria, o nível será reavaliado para verificar novos resultados.

Preparação para a Coleta de Dados

No primeiro nível de coleta, após o desenvolvimento das APIs, estas serão aplicadas em diversos microsserviços para extrair as métricas DORA de times *DevOps* específicos. Os resultados serão validados por meio de entrevistas com os membros dos times, verificando a coerência dos dados e sua contribuição para identificar melhorias nos microsserviços avaliados.

Além disso, o *DORA DevOps Quick Check* (RESEARCH; (DORA), 2023) será aplicado aos times da amostra para avaliar a compatibilidade entre a maturidade declarada e os resultados obtidos. Espera-se que melhores métricas providas pelas APIs estejam associadas a melhores resultados no *Quick Check*.

No segundo nível, serão avaliados os insumos essenciais de dados que alimentam as métricas, verificando se possuem qualidade mínima necessária para garantir a coerência dos serviços de mensuração. Também será analisada a disponibilidade desses dados em relação às ferramentas que os sustentam (bancos de dados, rotinas, *pipelines*, entre outros).

No terceiro nível, serão inspecionadas as documentações, manuais e roteiros corporativos utilizados pelos times *DevOps*. Será avaliado o quanto essas informações contribuem para a transformação digital e facilitam a evolução do desempenho *DevOps*.

Critérios de Qualidade

- **Validade de construto:** será avaliado se as métricas DORA estão sendo corretamente interpretadas e mensuradas. A validação ocorrerá após a disponibilização das APIs, que retornarão o nível de desempenho *DevOps* conforme a Tabela 2.2, classificando os resultados em baixo, médio ou alto.
- **Validade interna:** será verificado se os resultados obtidos decorrem efetivamente dos dados propostos. - Na métrica de frequência de implantações, serão contabilizadas as quantidades de *deploys* realizados em cada microsserviço. - No tempo gasto para mudanças, serão registrados os momentos (*timestamps*²²) da alteração

²²Representação numérica de uma data utilizada em linguagens de programação e bancos de dados.

do estado da história de usuário de “em andamento” até “pronto”. - No tempo de recuperação, serão registrados os *timestamps* de abertura e fechamento dos incidentes. - Na taxa de falhas, será contabilizada a quantidade de incidentes abertos para cada microsserviço.

- **Validade externa:** os resultados serão conferidos junto ao público-alvo, isto é, os integrantes dos times *DevOps*, verificando se os dados refletem de forma coerente a realidade dos microsserviços sustentados.
- **Confiabilidade:** os resultados das métricas DORA serão validados em diferentes microsserviços, permitindo a conferência da interpretação dos dados. A comparação será mais consistente quando aplicada a microsserviços com arquiteturas e volumes de consumo distintos, de modo a verificar se existem tendências de resultados independentemente do microsserviço avaliado.

Ameaças à Validade

Na análise do fluxo da *pipeline* existente foram identificadas as ferramentas responsáveis por fornecer os dados necessários ao cálculo das métricas DORA. Os dados específicos refletem de forma coerente o comportamento real dos microsserviços, assegurando a qualidade dos resultados obtidos.

4.7.2 DORA Quick Check

Para validar a eficácia do modelo criado, foi realizada em maio de 2024 uma avaliação de desempenho *DevOps* dos times envolvidos utilizando o *DORA Quick Check* (RESEARCH; (DORA), 2023). Essa avaliação preliminar estabelece uma linha de base que será comparada com resultados futuros, obtidos pelas APIs de métricas de velocidade e pela extração manual das métricas de estabilidade.

A Tabela 4.7 sintetiza os resultados obtidos. Os dados revelam que o tempo médio para mover o código até produção varia entre uma semana e um mês, com frequência de implantações semelhante. Aproximadamente 30,6% das alterações resultam em degradação de serviço, e o tempo para restaurar falhas é de um dia a uma semana. O resultado global foi de **4.1 em 10 pontos**, posicionando os times abaixo da média do setor financeiro e de outras indústrias.

Segundo a pesquisa DORA (*DevOps Research and Assessment*), mencionada na seção 2.3, diversas capacidades-chave impulsionam maior entrega de *software* e melhor desempenho organizacional (RESEARCH; (DORA), 2023). Entre elas, destacam-se três dimensões avaliadas: integração contínua, arquitetura desacoplada e cultura organizacional generativa.

Aspecto Avaliado	Resultado
Tempo médio para mover o código desde a confirmação até produção	Uma semana a um mês
frequência de implantações em produção	Uma vez por semana a uma vez por mês
Porcentagem de alterações com degradação de serviço	30,6%
Tempo para restaurar serviço após falha em produção	Um dia a uma semana
Resultado global no DORA Quick Check	4.1 de 10
Integração Contínua (CI)	9.0 de 10
Arquitetura Desacoplada	5.3 de 10
Cultura Organizacional Generativa	6.9 de 10

Tabela 4.7: Resumo da avaliação de desempenho *DevOps* pelo DORA Quick Check

A prática de integração contínua (CI) obteve **9.0 em 10 pontos**, evidenciando um processo automatizado robusto de construção. Esse resultado positivo decorre da existência de *scripts* que permitem a criação de pacotes implantáveis em qualquer ambiente.

A arquitetura desacoplada recebeu **5,3 em 10 pontos**, refletindo a elevada dependência entre sistemas. Mudanças em larga escala impactam diretamente outros times, restringindo a autonomia para implantações rápidas.

Por fim, a cultura organizacional, avaliada segundo a tipologia de Westrum (WESTRUM, 2004), alcançou **6.9 em 10 pontos**. A confiança mútua e o bom fluxo de informações observados nos times contribuem para uma cultura generativa, que favorece a cooperação e melhora o desempenho na entrega de *software*.

4.7.3 Avaliação das APIs

Na amostra foram considerados os microsserviços que sustentam o processo de originação de crédito, um dos serviços essenciais da instituição analisada. Esse serviço é mantido por dois times *DevOps* habilitadores: o primeiro responsável pelo desenvolvimento e versionamento das funcionalidades do produto, e o segundo voltado à evolução tecnológica e arquitetural da solução.

A originação de crédito compreende todas as etapas, desde a solicitação de um empréstimo até o desembolso dos fundos. Trata-se do processo que mobiliza o cliente a contratar um novo empréstimo, abrangendo a análise de perfil, a oferta de simulações de planos, até a formalização contratual. Após a contratação, o empréstimo segue para o fluxo de condução do crédito, não contemplado nesta amostra.

Considerando todos os microsserviços do ecossistema de originação de crédito, foi identificada uma frequência média de 16,2 implantações por mês. O tempo médio para mudanças foi de 9 dias, enquanto o tempo médio para recuperação foi de 2 dias. A taxa média de falhas nas alterações correspondeu a 30,6%. A análise da distribuição de

cada métrica revelou variação significativa entre os percentis superiores (25%) e inferiores (25%) dos microsserviços avaliados.

No contexto desta pesquisa, foram consideradas as métricas DORA, sua taxa de adoção e as organizações que as utilizam. Essas organizações foram analisadas para fornecer uma visão de referência sobre práticas *DevOps* e os requisitos necessários para alcançar desempenho elevado. A partir dos resultados, constatou-se que, conscientes do funcionamento das métricas, os times devem considerar, para fins de desempenho *DevOps*:

- Avaliação da produtividade das mudanças;
- Contagem de *hotfixes* em mensagens de *commit*;
- Quantidade de reversões em relação ao número de implantações bem-sucedidas;
- Percentual de liberações realizadas por meio de “liberação de correção”;
- Identificação de falhas com base em dados de monitoramento.

4.7.4 Investigação com Grupos Focais

Participantes

Os grupos focais foram compostos por 28 participantes, distribuídos em três perfis distintos: 12 desenvolvedores, 7 operadores de infraestrutura e operações de TI e 9 gestores de tecnologia e líderes de times *DevOps*.

A seleção seguiu o critério de representatividade funcional, priorizando profissionais diretamente envolvidos nos processos de desenvolvimento, operação e gestão.

Foram incluídos participantes com experiência mínima de dois anos em práticas *DevOps* na instituição, assegurando familiaridade com os fluxos de trabalho e relevância das percepções coletadas.

Coleta de Dados

A coleta foi realizada por meio de entrevistas exploratórias em formato de grupos focais, conduzidas em agosto de 2024 na sede da instituição. Cada sessão teve duração aproximada de 90 minutos e foi organizada de acordo com o perfil dos participantes.

As perguntas foram adaptadas ao papel de cada grupo, mas seguiram uma linha de investigação comum, centrada na efetividade das métricas DORA e no impacto das práticas *DevOps* sobre produtividade, estabilidade e tomada de decisão.

As entrevistas foram transcritas integralmente e, para fins de análise, trechos selecionados são apresentados na seção de resultados como evidências empíricas.

Análise dos Dados

Os depoimentos foram submetidos a codificação temática, identificando categorias recorrentes como produtividade, burocracia, estabilidade, integração entre times e visibilidade gerencial. A análise qualitativa seguiu abordagem indutiva, permitindo que os temas emergissem diretamente das falas dos participantes. Posteriormente, os resultados foram triangulados com os dados quantitativos das métricas DORA e com a literatura especializada, de modo a integrar evidências técnicas, culturais e organizacionais.

Validade e Confiabilidade

Para assegurar validade e confiabilidade, foram adotados três procedimentos principais: (i) triangulação de fontes, integrando métricas quantitativas, entrevistas qualitativas e referências bibliográficas; (ii) codificação independente por dois pesquisadores, reduzindo vieses individuais na interpretação dos dados; e (iii) registro sistemático das sessões, garantindo rastreabilidade das evidências. A escolha dos grupos focais como ferramenta de coleta foi justificada pela necessidade de captar percepções coletivas e individuais em ambiente colaborativo, favorecendo a emergência de diferentes perspectivas sobre o uso das métricas DORA.

4.7.5 Percepções sobre o Uso das Métricas DORA

Para a realização dos grupos focais, foram organizadas três sessões, cada uma com um público-alvo distinto — desenvolvedores, operadores de TI e gerentes — com o objetivo de obter uma compreensão abrangente sobre o uso das métricas DORA e das práticas *DevOps* na organização. Cada sessão teve duração aproximada de 90 minutos e buscou identificar percepções, práticas e desafios específicos de cada perfil.²³

A técnica de entrevista exploratória, descrita na seção 3.5.1, foi utilizada para melhor atender às necessidades dos grupos focais. As perguntas foram adaptadas conforme o papel dos participantes, mas seguiam uma linha de investigação comum, especialmente quanto à efetividade e ao impacto das métricas DORA nas operações e na colaboração entre times.

No grupo de desenvolvedores, foram discutidas questões relacionadas à produtividade, frequência de implantações e colaboração com operadores. Os operadores de TI, por sua vez, abordaram aspectos de estabilidade do sistema, tempo de recuperação de falhas e integração com desenvolvedores. Os gerentes concentraram-se na visibilidade para decisões estratégicas, entrega de valor e qualidade das mudanças.

²³As entrevistas nos grupos focais foram realizadas em Brasília, no edifício-sede da instituição, em agosto de 2024.

Percepções por Perfil

- **Desenvolvedores:**

- **Produtividade e eficiência:** 5 desenvolvedores relataram que as métricas DORA aumentaram a visibilidade sobre as entregas. No entanto, 7 mencionaram que o processo de implementação de mudanças permanece burocrático e ineficiente. Conforme destacou o Desenvolvedor D:

“Agora temos métricas, mas as práticas e a cultura ainda não mudaram de forma significativa.”

- **Frequência de implantações:** Apenas três desenvolvedores relataram aumento na frequência de entregas, enquanto os demais apontaram que os ciclos de revisão e aprovação permanecem demorados. O Desenvolvedor F comentou:

“Ainda precisamos esperar semanas para que um deploy simples seja aprovado.”

- **Colaboração entre times:** 6 desenvolvedores relataram dificuldades na comunicação com os operadores de TI, resultando em retrabalho e atrasos. O Desenvolvedor H afirmou:

“Passamos mais tempo esperando ajustes dos operadores do que avançando nos projetos.”

- **Operadores de TI:**

- **Estabilidade e recuperação de falhas:** 4 operadores relataram que o tempo médio de recuperação de falhas não apresentou melhorias significativas. O Operador C destacou:

“As métricas estão aí, mas as respostas continuam demoradas.”

- **Integração com desenvolvedores:** 5 operadores expressaram frustração com o foco dos desenvolvedores em entregas rápidas, o que compromete a estabilidade do sistema. Conforme o Operador E:

“Parece que somos os únicos preocupados com a qualidade das entregas.”

- **Monitoramento e proatividade:** Apenas 2 operadores perceberam melhorias no monitoramento contínuo. O Operador G comentou:

“Conseguimos identificar problemas, mas ainda precisamos de mais recursos para agir rapidamente.”

- **Gerentes:**

- **Visibilidade e decisões estratégicas:** 5 gerentes relataram que a visibilidade proporcionada pelas métricas DORA ainda é insuficiente para decisões estratégicas em tempo real. O Gerente A mencionou:

“As métricas mostram apenas parte do que realmente precisamos para tomar decisões estratégicas.”

- **Melhoria na entrega de valor:** 3 gerentes notaram alguma melhoria no tempo gasto para mudanças, mas consideraram os avanços marginais. O Gerente C explicou:

“As melhorias são pontuais, ainda temos um longo caminho.”

- **Foco na qualidade:** 2 gerentes observaram redução na taxa de falhas, mas ainda a consideraram elevada. O Gerente F comentou:

“Precisamos reduzir a taxa de falhas de forma mais consistente.”

4.7.6 Integração dos Resultados Quantitativos e Qualitativos

A aplicação da metodologia empírica possibilitou a articulação entre os dados quantitativos, obtidos pelas métricas DORA, e as evidências qualitativas provenientes dos grupos focais. Essa integração permitiu compreender não apenas os valores numéricos, mas também os significados atribuídos pelos diferentes perfis de participantes.

A frequência média de 16,2 implantações mensais e o tempo médio de entrega para mudanças, de 9 dias, foram interpretados pelos desenvolvedores como insuficientes diante da burocracia persistente nos processos de aprovação. De modo semelhante, a taxa de falhas de 30,6% foi associada pelos operadores à ausência de automação e às dificuldades de integração com os desenvolvedores. Os gestores, por sua vez, reconheceram avanços pontuais, mas destacaram que os indicadores ainda não oferecem suporte adequado para decisões estratégicas em tempo real.

Esses resultados evidenciam que os indicadores quantitativos não podem ser analisados de forma isolada, devendo ser interpretados em conjunto com as percepções culturais e organizacionais. A triangulação entre métricas, entrevistas e literatura especializada reforça a validade interna e externa da pesquisa, demonstrando que a maturidade *DevOps* depende simultaneamente de avanços técnicos, de práticas estruturais e de transformações culturais.

4.7.7 Síntese das Percepções dos Grupos Focais

Os grupos focais apontaram que a adoção das métricas DORA apresentou impacto inferior ao esperado. Desenvolvedores enfrentam processos burocráticos, operadores lidam com falhas recorrentes e ausência de automação, enquanto gerentes encontram dificuldades em utilizar as métricas para decisões estratégicas. O desempenho *DevOps* da organização permanece aquém das expectativas, com pontuação geral de **4,1 em 10** no *DORA Quick Check*, conforme discutido na seção de resultados. Para ampliar o impacto das métricas DORA, torna-se necessário alinhar práticas entre os times, fomentar uma cultura colaborativa e investir em ferramentas de automação.

4.7.8 Comparação do Uso das Métricas DORA

Com o objetivo de avaliar os impactos da adoção das métricas DORA, foi realizada uma comparação entre a situação anterior à mensuração sistemática e os resultados observados após a implementação. Essa análise foi complementada por percepções qualitativas obtidas em entrevistas com especialistas, permitindo identificar mudanças práticas e culturais nos times.

Métrica DORA	Antes	Depois	Evidência qualitativa
Frequência de implantações	Implantações esporádicas, sem padrão	Implantações semanais ou diárias, mais previsíveis	Maior confiança na cadência de entregas
Tempo gasto para mudanças	Prazo médio de semanas a meses	Prazo reduzido para dias ou poucas semanas	Percepção de maior agilidade e resposta rápida às demandas
Taxa de falhas	Incidentes pouco monitorados, sem indicador consolidado	Taxa mensurada e acompanhada em cada ciclo	Uso dos dados para prevenção de falhas
Tempo médio de recuperação	Correções demoradas, sem métrica formal	Correções monitoradas, tempo médio reduzido	Evolução para abordagem preditiva

Tabela 4.8: Situação antes e depois da adoção das métricas DORA

A Tabela 4.8 evidencia que a adoção das métricas DORA promoveu uma transformação significativa nos processos de desenvolvimento e operação. Antes da mensuração, as práticas eram pouco estruturadas, caracterizadas por baixa previsibilidade e ausência de indicadores claros. Após a implementação, verificou-se maior regularidade nas entregas, redução dos prazos de mudança, acompanhamento sistemático das falhas e melhoria na capacidade de recuperação.

As entrevistas confirmaram que os times passaram a utilizar os dados não apenas para monitorar desempenho, mas também para antecipar problemas e orientar decisões estratégicas. Essa evolução reforça o papel das métricas DORA como catalisadoras da maturidade organizacional e cultural no contexto *DevOps*.

Capítulo 5

Discussão de Resultados

O ponto central desta análise não são as métricas em si, mas a evolução dos times em relação ao *DevOps*. As métricas DORA, neste contexto, funcionam como instrumentos para mensurar o progresso rumo a metas de eficiência, fornecendo indicadores que ajudam os membros técnicos a identificar oportunidades de melhoria com ênfase na qualidade das entregas.

Neste trabalho, o desenvolvimento concentrou-se nas métricas de velocidade (frequência de implantações e tempo de entrega para mudanças), cuja implementação demandou esforço significativo de integração entre ferramentas, formalização de algoritmos e validação de dados. Os resultados obtidos demonstram impacto direto sobre o desempenho observado nos times.

Coerência metodológica e técnica

A escassez de estudos sobre métricas DORA aplicadas a instituições financeiras de grande porte justificou a adoção de uma abordagem metodológica adaptativa, combinando revisão *ad hoc* e *snowballing*. A formalização dos algoritmos em lógica de primeira ordem garantiu rastreabilidade e consistência nos cálculos, especialmente relevantes em ambientes regulados.

Os algoritmos para frequência de implantações e tempo de entrega para mudanças foram implementados em arquitetura MVC com APIs REST, integrando GitLab, Jenkins, JFrog e Sonar, o que permitiu extração automatizada confiável e repetível das métricas de velocidade. Por outro lado, as métricas de estabilidade (tempo médio de recuperação e taxa de falhas) não foram desenvolvidas devido à ausência de integração entre a esteira de desenvolvimento e a ferramenta de gestão de incidentes, restringindo o escopo às dimensões de velocidade.

Desempenho atual e barreiras práticas

A avaliação preliminar do desempenho *DevOps* por meio do *DORA Quick Check* evidenciou avanços técnicos e limitações estruturais: resultado global de 4,1 em 10; integração contínua com 9,0 em 10, indicando automação robusta; arquitetura desacoplada com 5,3 em 10, refletindo dependências entre sistemas; e cultura organizacional com 6,9 em 10, sugerindo fluxo de informações razoável, porém insuficiente para decisões estratégicas.

As métricas efetivamente desenvolvidas indicam:

- Frequência de implantações: média de 16,2 entregas mensais, com variação significativa entre serviços. Essa métrica evidenciou padrões de cadência e gargalos em serviços mais acoplados.
- Tempo de entrega para mudanças: média de 9 dias, com predominância da etapa de implantação no ciclo. A decomposição do tempo mostrou que a fase de operações concentra o maior tempo acumulado, indicando oportunidades de otimização nos fluxos de liberação.

A Tabela 4.8 sintetiza o cenário antes e depois da mensuração sistemática: maior regularidade nas entregas e prazos de mudança reduzidos. Ainda assim, a evolução permanece desigual e vulnerável, limitada por processos de liberação com múltiplas etapas de aprovação manuais e por arquitetura ainda acoplada.

Evidências qualitativas

As entrevistas e grupos focais revelaram que, apesar da introdução das métricas, a transformação cultural e organizacional permanece incompleta. Os depoimentos indicam que os desenvolvedores percebem maior visibilidade, mas pouca mudança prática; operadores enfatizam a necessidade de maior automação e integração; e gerentes reconhecem avanços pontuais, porém insuficientes para subsidiar decisões estratégicas em tempo real.

Essas percepções reforçam que as métricas DORA, quando consideradas isoladamente, não asseguram a melhoria contínua. Para que cumpram plenamente seu papel, é necessário que sejam acompanhadas de mudanças nos processos de aprovação, maior autonomia dos times e práticas de monitoramento proativo. A análise qualitativa, portanto, complementa os resultados numéricos ao evidenciar que os gargalos não se restringem à dimensão técnica, mas envolvem também aspectos culturais e organizacionais.

No âmbito do desenvolvimento, os participantes destacaram a burocracia nos processos de aprovação e a comunicação deficiente com operações, fatores que geram retrabalho e atrasos de liberação. Como relatou um dos entrevistados:

“Agora temos métricas, mas as práticas e a cultura ainda não mudaram de forma significativa.” (Desenvolvedor D, grupo focal, 2024)

Na infraestrutura, foi ressaltado o potencial das métricas para prever falhas a partir de padrões históricos, embora se reconheça a necessidade de maior automação e proatividade. Um operador sintetizou essa percepção ao afirmar:

“As métricas estão aí, mas as respostas continuam demoradas.” (Operador C, grupo focal, 2024)

Por sua vez, na área de monitoração, observou-se a expectativa de transição de um modelo reativo para um modelo preditivo, dependente de integração de dados e maior observabilidade. Um especialista destacou:

“As métricas DORA representam uma oportunidade de evolução para o monitoramento, permitindo antecipar falhas com base em padrões históricos.” (Especialista de monitoração, entrevista pessoal, 2024)

De forma geral, a cultura organizacional foi descrita como colaborativa e orientada a ciclos curtos, mas ainda marcada por dependências entre times e autonomia limitada para implantações independentes. Esse cenário reforça a relevância de evoluir os indicadores de estabilidade, ainda ausentes no modelo atual, como condição para consolidar avanços técnicos e culturais de forma integrada.

Contexto organizacional e escala

O contexto de atuação da instituição, seja *business-to-business* (B2B) ou *business-to-customer* (B2C), influencia diretamente a adoção e o valor das capacidades DORA. Ambientes B2C tendem a se beneficiar de ciclos acelerados de implementação, enquanto ambientes B2B enfrentam restrições adicionais, como prazos contratuais, requisitos regulatórios e maior responsabilidade sobre falhas, com custos mais elevados.

A escala institucional e a diversidade de plataformas de TI impactam a introdução dessas práticas. Em organizações de menor porte, a implementação das capacidades DORA é, em geral, mais simples e direta; em instituições de grande porte, uma abordagem *top-down* costuma ser ineficaz. É essencial que os times tenham autonomia para decidir como e quando implementar práticas *DevOps*, com apoio integral da gestão, pois muitas capacidades dependem de mudanças culturais que emergem organicamente dos times.

Implicações estratégicas e técnicas

Os resultados mostram avanços mensuráveis nas dimensões de velocidade, condicionados por fatores estruturais. Para consolidar a evolução, recomenda-se:

- Governança de mudanças: simplificar fluxos de aprovação e adotar portões inteligentes, reduzindo o tempo de entrega para mudanças.

- Arquitetura: modularização, contratos estáveis e uso de *feature flags*, ampliando a autonomia dos times para implantações independentes.
- Qualidade contínua: fortalecer testes de integração e contrato, adotar cobertura orientada a risco e mecanismos de rollback automatizado.
- Dados e mensuração: evoluir de APIs para *logs* com normalização e governança, viabilizando a futura mensuração do tempo médio de recuperação e da taxa de falhas.

Limitações e agenda de evolução

A principal limitação foi a ausência de automação das métricas de estabilidade, dependentes de coleta manual. A heterogeneidade de ferramentas e processos dificultou a visão unificada das métricas e explica a inviabilidade de integrar incidentes à esteira de desenvolvimento. Os resultados estão restritos ao domínio analisado e devem ser aplicados em outras áreas para validação externa.

A assimetria entre serviços exige trilhas de evolução específicas, com metas alinhadas ao contexto de cada time. A consolidação das métricas como instrumento de gestão requer patrocínio, alinhamento entre áreas e incentivos que valorizem estabilidade junto com velocidade.

Síntese final

O modelo proposto demonstrou solidez técnica e gerou avanços mensuráveis nas métricas de velocidade, mostrando ser possível aumentar a cadência e reduzir prazos de forma consistente. Por outro lado, a ausência das métricas de estabilidade indica que os ganhos ainda são parciais, pois não contemplam plenamente a confiabilidade operacional.

Para que os avanços em velocidade se convertam em impacto organizacional duradouro, é necessário evoluir simultaneamente em processos (governança de mudanças menos burocrática), arquitetura (maior desacoplamento e autonomia de implantação) e dados (integração com logs e ferramentas de incidentes). Com essas frentes, a instituição pode migrar de uma maturidade fragmentada para ciclos de entrega mais curtos, previsíveis e confiáveis, alinhando velocidade e qualidade na geração de valor.

Capítulo 6

Considerações Finais

Este trabalho teve como objetivo avaliar o desempenho *DevOps* em uma instituição financeira de grande porte, a partir da aplicação das métricas DORA e da formalização de um modelo técnico-metodológico para mensuração.

A pesquisa concentrou-se na construção de um método adaptado ao contexto regulado, na implementação de algoritmos para cálculo das métricas de velocidade (frequência de implantações e tempo de entrega de mudanças) e na análise integrada de evidências quantitativas, qualitativas e contextuais.

6.1 Síntese dos Resultados

A investigação demonstrou que é possível estruturar um modelo de mensuração do desempenho *DevOps* em ambientes complexos e regulados, conciliando rigor técnico e aplicabilidade prática.

Os resultados evidenciaram avanços mensuráveis nas métricas de velocidade, com maior regularidade nas entregas e redução dos prazos de mudança. As entrevistas com especialistas reforçaram que os desafios não se limitam à dimensão técnica, mas envolvem barreiras culturais, organizacionais e de governança.

A análise comparativa antes e depois da mensuração mostrou que a adoção sistemática das métricas contribuiu para ampliar a previsibilidade e a capacidade de gestão dos processos de entrega.

6.2 Contribuições

Contribuições científicas

As principais contribuições deste trabalho podem ser sintetizadas em três dimensões:

- **Modelo:** formalização de um modelo de mensuração do desempenho *DevOps* baseado em métricas DORA, adaptado ao contexto de instituições financeiras de grande porte, com algoritmos definidos em lógica de primeira ordem para garantir rastreabilidade e reprodutibilidade.
- **Método:** desenvolvimento de uma estratégia metodológica inédita, combinando revisão *ad hoc*, *snowballing*, integração de dados por meio de APIs e entrevistas qualitativas, permitindo articular evidências técnicas e organizacionais.
- **Análise:** realização de uma investigação inédita sobre a aplicação das métricas DORA em ambientes financeiros regulados, destacando limitações, barreiras e implicações práticas para a evolução do desempenho *DevOps*.

Benefícios para as partes interessadas

O método e o artefato desenvolvidos oferecem benefícios concretos para diferentes públicos:

- **Times de desenvolvimento:** maior clareza sobre o ritmo das entregas e identificação de gargalos, suporte à tomada de decisão técnica com base em dados objetivos, redução de incertezas sobre prazos e maior autonomia para planejar implantações.
- **Gestores:** visibilidade consolidada sobre o desempenho dos times, capacidade de identificar áreas críticas e priorizar investimentos, suporte à governança de mudanças com indicadores confiáveis e alinhamento entre velocidade e qualidade.
- **Organização:** fortalecimento da cultura orientada a dados, aumento da previsibilidade operacional, redução de riscos associados a falhas e atrasos, e criação de uma base sólida para a evolução do desempenho *DevOps* em escala institucional.

6.3 Limitações

Apesar dos avanços, algumas limitações devem ser reconhecidas:

- As métricas de estabilidade (tempo médio de recuperação e taxa de falhas) não foram desenvolvidas devido à ausência de integração entre a esteira de desenvolvi-

mento e a ferramenta de gestão de incidentes, restringindo a análise às dimensões de velocidade.

- A generalização dos resultados está restrita ao domínio de crédito, sendo necessária a aplicação do modelo em outros setores da instituição para validação externa.
- A heterogeneidade de ferramentas e processos dificultou a obtenção de uma visão unificada, exigindo esforços adicionais de padronização e governança.

6.4 Conclusão

Este trabalho demonstrou que é possível mensurar o desempenho *DevOps* em instituições financeiras de grande porte por meio das métricas DORA, mesmo em ambientes regulados e complexos. A implementação do método e do artefato resultou em avanços concretos nas métricas de velocidade, ampliando a previsibilidade das entregas e fortalecendo a capacidade de gestão. Além disso, o estudo forneceu evidências científicas e práticas que podem orientar a evolução do desempenho *DevOps* em escala institucional.

Do ponto de vista científico, a pesquisa contribui em três frentes complementares: (i) a proposição de um **modelo** de mensuração do desempenho *DevOps* adaptado ao contexto regulado de instituições financeiras de grande porte; (ii) a elaboração de um **método** técnico-metodológico inédito, que integra evidências quantitativas e qualitativas; e (iii) a condução de uma **análise inédita** sobre a aplicação das métricas DORA em ambientes regulados. Essas contribuições reforçam o caráter original da investigação e oferecem fundamentos sólidos para futuras pesquisas e citações acadêmicas, estabelecendo uma base para que organizações possam migrar de uma maturidade fragmentada para ciclos de entrega mais curtos, confiáveis e sustentáveis.

6.5 Trabalhos Futuros

A continuidade desta pesquisa pode seguir diferentes direções:

- Evolução da coleta de dados para *logs*, permitindo maior granularidade e viabilizando a mensuração das métricas de estabilidade.
- Expansão da aplicação do modelo para outros domínios da instituição, como pagamentos, investimentos e canais digitais.
- Integração das métricas DORA com indicadores de negócio, como satisfação do cliente, tempo de resposta a demandas regulatórias e eficiência operacional.

- Investigação sobre o impacto cultural da adoção das métricas, explorando como autonomia, colaboração e alinhamento entre áreas influenciam a maturidade *DevOps*.

Referências Bibliográficas

- ALUNNI, A. *Innovation Finance and Technology Transfer: Funding Proof-of-Concept*. 1. ed. London: Routledge, 2019. ISBN 9780367331580. Disponível em: <<https://www.routledge.com/Innovation-Finance-and-Technology-Transfer-Funding-Proof-of-Concept/Alunni/p/book/9780367331580>>. 50
- AMARO, R.; PEREIRA, R.; SILVA, M. M. da. Capabilities and practices in devops: A multivocal literature review. *IEEE Transactions on Software Engineering*, v. 49, n. 2, p. 883–901, 2023. 28, 29
- ARTAC, M. et al. Devops: Introducing infrastructure-as-code. In: *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. Buenos Aires, Argentina: [s.n.], 2017. p. 497–498. 30
- Atlassian. *DORA Metrics: How to Measure Open DevOps Success*. 2025. Atlassian Blog. Guia prático sobre aplicação das métricas DORA em ambientes Open DevOps. Acesso em: 07 dez. 2025. Disponível em: <<https://www.atlassian.com/devops/frameworks/dora-metrics>>. 34
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. 3. ed. Boston: Addison-Wesley, 2012. 47, 72
- BEZEMER, C.-P. et al. How is performance addressed in devops? In: *Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering*. Mumbai, India: [s.n.], 2019. p. 45–50. 32
- BIRD, J. *DevOps for Finance*. 2015. O'Reilly Media (informação catalogada no Google Books). Disponível em: <<https://books.google.com.br/books?id=J20-zQEACAAJ>>. 5, 6
- BOHANEC, P.; MORELLI, F. Improving software delivery and operational performance with dora (devops research and assessment) best practices. *Anwendungen und Konzepte der Wirtschaftsinformatik*, v. 1, n. 17, p. 3, 2023. 28
- BOOCH, G. *UML: guia do usuário*. 2. ed. Rio de Janeiro: Elsevier Brasil, 2006. Tradução da obra original *UML User Guide*. ISBN 9788535214729. 48, 49, 50
- BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, v. 80, n. 4, p. 571–583, 2007. 37

CAPES, Portal Periódicos. *Portal Periódicos CAPES*. 2020. Acesso em: 07 dez. 2025. Disponível em: <<http://www.periodicos.capes.gov.br>>. 60

CARTER, S. et al. *Model-Driven DevOps: Increasing Agility and Security in Your Physical Network Through DevOps*. 1. ed. Boston, MA: Addison-Wesley Professional, 2022. ISBN 9780137371867. Disponível em: <<https://www.pearson.com/en-us/subject-catalog/p/model-driven-devops/P200000007465>>. 13

CELLARD, A. A pesquisa qualitativa: enfoques epistemológicos e metodológicos. *Revista de Administração da UFSM*, v. 3, n. 2, p. 380–405, 2010. ISSN 1983-4659. 39

Checkmarx. *Implementing DORA Metrics: Best Practices for 2025*. 2025. Checkmarx Blog. Discussão sobre como reduzir gargalos e aumentar confiabilidade com DORA. Acesso em: 08 dez. 2025. 34

CYSNEIROS, L. M. *Requisitos não funcionais: da elicitação ao modelo conceitual*. Tese (Tese (Doutorado em Informática)) — Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001. 47, 72

DENZIN, N. K.; LINCOLN, Y. S. *The Sage Handbook of Qualitative Research*. 4. ed. Thousand Oaks, CA: Sage Publications, 2011. ISBN 9781412974173. 54

EASTERBROOK, S. et al. Selecting empirical methods for software engineering research. In: *Guide to Advanced Empirical Software Engineering*. London: Springer, 2008. p. 285–311. ISBN 9781848000445. 56

EBERT, C.; HOCHSTEIN, L. Devops in practice. *IEEE Software*, v. 40, n. 1, p. 29–36, 2022. 26, 27, 28

EITAN, A. T. et al. *Connected Papers: Literature Search Workflows*. 2024. Acesso em: 05 jan. 2024. Disponível em: <<https://www.connectedpapers.com/main/b0855f2715d00f500ac72f1aa4b7bbb13f841401/DevOps-Metrics/graph>>. 60

ERICH, F. M. A.; AMRIT, C.; DANEVA, M. A qualitative study of devops usage in practice. *Journal of Software: Evolution and Process*, v. 29, n. 6, p. e1885, 2017. Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1002/smr.1885>>. 30

FAJARDO, R. A. d. S. *Lógica matemática*. 1. ed. São Paulo: Edusp, 2017. ISBN 9788531416462. 40

FITZGERALD, B.; STOL, K.-J. Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, v. 123, p. 176–189, 2017. 1

FORSGREN, N. *The Key to High Performance: What the Data Says*. [S.l.]: IT Revolution Press, 2018. Acesso em: 14 abr. 2023. 25

FORSGREN, N.; HUMBLE, J. Devops: Profiles in itsm performance and contributing factors. In: *Proceedings of the Western Decision Sciences Institute (WDSI)*. Las Vegas, NV: [s.n.], 2016. p. 1–6. 14

- FORSGREN, N.; HUMBLE, J. The role of continuous delivery in it and organizational performance. In: *Proceedings of the Western Decision Sciences Institute (WDSI)*. Las Vegas, NV: [s.n.], 2016. p. 1–6. 24
- FORSGREN, N.; HUMBLE, J.; KIM, G. *Accelerate State of DevOps 2018*. [S.l.], 2018. Relatório técnico. Disponível em: <<https://services.google.com/fh/files/misc/state-of-devops-2018.pdf>>. 2, 4, 6, 7, 11, 15, 17, 18, 19, 20, 31, 65
- FORSGREN, N.; KERSTEN, M. Devops metrics. *Communications of the ACM*, v. 61, n. 4, p. 44–48, 2018. 31
- FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison-Wesley, 2002. (Addison-Wesley Signature Series). ISBN 978-0321127426. 51, 52
- FOWLER, S. J. *Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization*. 1. ed. Sebastopol, CA: O’Reilly Media, 2016. ISBN 9781491965971. 30
- GAMMA, E. et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. [S.l.]: Addison-Wesley, 1995. (Addison-Wesley Professional Computing Series). ISBN 978-0201633610. 50, 52
- GAROUSI, V.; FELDERER, M.; MÄNTYLÄ, M. V. Guidelines for ad hoc and systematic reviews in software engineering. *Information and Software Technology*, v. 106, p. 82–100, 2019. 37
- GIBBS, A. Focus groups. *Social Research Update*, n. 36, p. 1–8, 2012. ISSN 1360-7898. Disponível em: <<https://sru.soc.surrey.ac.uk/SRU36.html>>. 44
- GODOY, A. S. Pesquisa qualitativa: tipos fundamentais. *Revista de Administração de Empresas*, v. 35, n. 3, p. 20–29, 1995. ISSN 0034-7590. 39
- GOMES, A. A. Estudo de caso - planejamento e métodos. *Nuances: Estudos sobre Educação*, v. 15, n. 16, p. 93–108, 2008. ISSN 1414-7106. 54
- Google Cloud / DORA. *DORA Research 2025 Overview*. [S.l.], 2025. Relatório oficial consolidando métricas DORA como padrão global. Acesso em: 07 dez. 2025. Disponível em: <<https://dora.dev/research/2025/>>. 34
- HAIGHT, C.; SPAFFORD, G. *Research Roundup for DevOps, 2022*. 2022. Acesso em: 07 dez. 2025. 1
- HAINDL, P.; PLÖSCH, R. Towards continuous quality: measuring and evaluating feature-dependent non-functional requirements in devops. In: *Proceedings of the 2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*. Gothenburg, Sweden: [s.n.], 2019. p. 91–94. 31
- HARALAYYA, B. Core banking technology and its top 6 implementation challenges. *Journal of Advanced Research in Operational and Marketing Management*, v. 4, n. 1, p. 25–27, 2021. 3

- HERING, M. *DevOps for the Modern Enterprise: Winning Practices to Transform Legacy IT Organizations*. 1. ed. Portland, OR: IT Revolution Press, 2018. ISBN 9781942788190. 25
- HUMBLE, J.; KIM, G. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. [S.l.]: IT Revolution, 2018. 17, 20
- HUMBLE, J.; MOLESKY, J.; O'REILLY, B. *Lean Enterprise: How High Performance Organizations Innovate at Scale*. 2. ed. Sebastopol, CA: O'Reilly Media, 2020. ISBN 9781492043074. 10, 12, 32
- JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. *The Unified Modeling Language Reference Manual*. 2. ed. Boston: Addison-Wesley, 2021. ISBN 9780321245625. 49
- JAYAKODY, V.; WIJAYANAYAKE, J. Critical success factors for devops adoption in information systems development. *International Journal of Information Systems and Project Management*, v. 11, n. 3, p. 65–82, 2023. ISSN 2182-7788. 28, 29
- JOHN, W. et al. Service provider devops. *IEEE Communications Magazine*, v. 55, n. 1, p. 204–211, 2017. 30
- JUNIOR, E. B. L. et al. Análise documental como percurso metodológico na pesquisa qualitativa. *Cadernos da FUCAMP*, v. 20, n. 44, p. 64–75, 2021. ISSN 1982-3860. 39
- KATAL, A.; BAJORIA, V.; DAHIYA, S. Devops: Bridging the gap between development and operations. In: *Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. Erode, India: [s.n.], 2019. p. 1–7. 31
- KIM, G. et al. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. 2. ed. Portland, OR: IT Revolution Press, 2021. 2, 4, 5, 6, 7, 11, 12, 14, 15, 19, 20, 33
- KOSE, B. O. Mobilizing devops: exploration of devops adoption in mobile software development. *Kybernetes*, v. 53, n. 7, 2024. Disponível em: <<https://doi.org/10.1108/K-04-2024-0989>>. 29
- KRUEGER, R. A.; CASEY, M. A. *Focus Groups: A Practical Guide for Applied Research*. 5. ed. Thousand Oaks, CA: Sage Publications, 2014. ISBN 9781483365244. 44, 45, 46
- KUNJUMOHAMED, S.; SATTARI, H. *Spring Essentials*. 1. ed. Birmingham: Packt Publishing, 2016. ISBN 9781785285177. 50
- LEE, R. P.; GREWAL, R. Strategic responses to new technologies and their impact on firm performance. *Journal of Marketing*, v. 68, n. 4, p. 157–171, 2004. 1
- LETHBRIDGE, T. C.; LAGANIERE, R. *Object-Oriented Software Engineering: Practical Software Development Using UML and Java*. 2. ed. New York: McGraw-Hill, 2005. ISBN 9780071240839. 52, 53, 54

- LETHBRIDGE, T. C.; SIM, S. E.; SINGER, J. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, v. 10, n. 3, p. 311–341, 2005. Disponível em: <<https://link.springer.com/article/10.1007/s10664-005-1290-x>>. 55, 56
- LWAKATARE, L. E. et al. Devops in practice: A multiple case study of five companies. *Information and Software Technology*, v. 114, p. 217–230, 2019. 26
- LÓPEZ-FERNÁNDEZ, D. et al. Devops team structures: Characterization and implications. *IEEE Transactions on Software Engineering*, v. 48, n. 10, p. 3716–3736, 2021. 32
- MAROUKIAN, K.; GULLIVER, S. R. *Leading DevOps practice and principle adoption*. 2020. ArXiv preprint arXiv:2008.10515. Acesso em: 11 mar. 2023. Disponível em: <<https://arxiv.org/abs/2008.10515>>. 32
- MAROUKIAN, K.; GULLIVER, S. R. The link between transformational and servant leadership in devops-oriented organizations. In: *Proceedings of the 2020 European Symposium on Software Engineering*. Athens, Greece: [s.n.], 2020. p. 21–29. 32
- MCCARTHY, M. A. et al. Composable devops: Automated ontology based devops maturity analysis. In: *Proceedings of the 2015 IEEE International Conference on Services Computing*. New York, USA: [s.n.], 2015. p. 600–607. 30
- MERRIAM, S. B.; TISDELL, E. J. *Qualitative Research: A Guide to Design and Implementation*. 4. ed. San Francisco, CA: John Wiley & Sons, 2015. ISBN 9781119003618. 35
- MORGAN, D. L. *Focus Groups as Qualitative Research*. 2. ed. Thousand Oaks, CA: Sage Publications, 1997. v. 16. (Qualitative Research Methods, v. 16). ISBN 9780761912535. 44
- PATTON, M. Q. *Qualitative Research & Evaluation Methods*. 3. ed. Thousand Oaks, CA: Sage Publications, 2002. ISBN 9780761919711. 35
- RAPLEY, T. J. The art (fulness) of open-ended interviewing: some considerations on analysing interviews. *Qualitative Research*, v. 1, n. 3, p. 303–323, 2001. 41, 42, 43, 44
- RESEARCH, D.; (DORA), A. *DORA DevOps Quick Check*. 2023. Ferramenta online de avaliação rápida de práticas DevOps. Disponível em: <<https://dora.dev/quickcheck/>>. 17, 82, 83
- ROCHE, J. Adopting devops practices in quality assurance: Merging the art and science of software development. *ACM Queue*, v. 11, n. 9, p. 20–27, 2013. 30
- ROSENTHAL, C.; JONES, N. *Chaos Engineering: System Resiliency in Practice*. [S.l.]: O’Reilly Media, 2020. 2, 5, 32
- RUIZ, J. M. S. et al. *A Benchmarking Proposal for DevOps Practices on Open Source Software Projects*. 2023. Acesso em: 11 abr. 2023. 33, 34, 60

- RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, v. 14, p. 131–164, 2009. Disponível em: <<https://link.springer.com/article/10.1007/s10664-008-9102-8>>. 55, 56, 57
- SALLIN, M. et al. Measuring software delivery performance using the four key metrics of devops. In: *Proceedings of the International Conference on Agile Software Development*. Trondheim, Norway: [s.n.], 2021. (Lecture Notes in Business Information Processing, v. 402), p. 103–119. 33
- SANJURJO, E. et al. Assessing bizdevops maturity using international standards: Case studies and lessons learned. *Journal of Software Evolution and Process*, v. 36, n. 8, p. e2646, 2024. Disponível em: <<https://doi.org/10.1002/smr.2646>>. 29
- SINGH, V. et al. Devops based migration aspects from legacy version control system to advanced distributed vcs for deploying micro-services. In: *Proceedings of the 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. Bangalore, India: [s.n.], 2021. p. 1–5. 33
- SMITH, D. et al. *Accelerate State of DevOps 2021*. Mountain View, CA, 2021. Relatório técnico. Disponível em: <<https://services.google.com/fh/files/misc/state-of-devops-2021.pdf>>. 21, 26
- SONI, M. *Hands-on Azure DevOps: CICD Implementation for Mobile, Hybrid, and Web Applications Using Azure DevOps and Microsoft Azure*. New Delhi: BPB Publications, 2020. 1. ed. 10, 18
- STATES, U. *Sarbanes–Oxley Act of 2002*. Washington, D.C.: [s.n.], 2002. Public Law 107-204, 116 Stat. 745. Acesso em: 09 fev. 2023. Disponível em: <<https://www.congress.gov/bill/107th-congress/house-bill/3763>>. 3
- STEWART, D. W.; SHAMDASANI, P. N. *Focus Groups: Theory and Practice*. 3. ed. Thousand Oaks, CA: Sage Publications, 2014. ISBN 9781412998070. 43, 44, 46
- TORBLE, T. Change management and devops: Back to the future. *ITNOW*, v. 61, n. 3, p. 52–53, 2019. 31
- TRIHINAS, D. et al. Devops as a service: Pushing the boundaries of microservice adoption. *IEEE Internet Computing*, v. 22, n. 3, p. 65–71, 2018. 31
- TURNER, J. R.; MCALLISTER, K. J. *Group Dynamics in Development: A Focus Group Methodology*. 1. ed. Cham: Springer, 2017. ISBN 9783319548022. 45
- VIERO, A. A completude da lógica de primeira ordem e o problema da existência matemática. *O que nos faz pensar*, v. 12, n. 15, p. 57–72, 2002. ISSN 0104-6675. 40
- WESTRUM, R. A typology of organisational cultures. *BMJ Quality & Safety*, v. 13, n. suppl 2, p. ii22–ii27, 2004. Disponível em: <https://qualitysafety.bmj.com/content/13/suppl_2/ii22>. 84
- WIEDEMANN, A. et al. The devops phenomenon. *ACM Queue*, v. 17, n. 2, p. 40–59, 2019. 25

WIEDEMANN, A. et al. Integrating development and operations teams: A control approach for devops. *Information and Organization*, Elsevier, v. 33, n. 3, p. 100474, 2023. 1

WIENER, R.; PINSON, L. J. *Fundamentals of OOP and Data Structures in Java*. 1. ed. Cambridge: Cambridge University Press, 2000. ISBN 9780521644004. 52, 53

WILKES, B.; MILANI, A. M. P.; STOREY, M.-A. (dora) metrics. In: *Proceedings of the 2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. Bogotá, Colombia: [s.n.], 2023. p. 62–72. 33, 34

WINTERS, T.; MANSHRECK, T.; WRIGHT, H. *Software Engineering at Google: Lessons Learned from Programming over Time*. 1. ed. Sebastopol, CA: O'Reilly Media, 2020. ISBN 9781492082798. Disponível em: <<https://www.oreilly.com/library/view/software-engineering-at/9781492082781/>>. 10

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE)*. ACM, 2014. p. 321–330. Disponível em: <<https://doi.org/10.1145/2601248.2601268>>. 38, 59

YIN, R. K. *Case Study Research: Design and Methods*. 4. ed. Thousand Oaks, CA: Sage Publications, 2009. ISBN 9781412960992. 54

ŠĆEKIĆ, M. et al. Application of devops approach in developing business intelligence system in bank. In: *Proceedings of the 2018 7th Mediterranean Conference on Embedded Computing (MECO)*. Budva, Montenegro: [s.n.], 2018. p. 1–4. 31