



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RAYAN VICTOR DE SOUZA FREITAS

**ADOÇÃO DE DEVOPS: PRÁTICAS, DESAFIOS E RECOMENDAÇÕES PARA A
MELHORIA NA IMPLEMENTAÇÃO**

RUSSAS

2025

RAYAN VICTOR DE SOUZA FREITAS

ADOÇÃO DE DEVOPS: PRÁTICAS, DESAFIOS E RECOMENDAÇÕES PARA A
MELHORIA NA IMPLEMENTAÇÃO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção
do grau de Bacharel em Ciência da Computação.

Orientadora: Prof. Ms. Valéria Maria da
Silva Pinheiro.

RUSSAS

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

F938a Freitas, Rayan.

ADOÇÃO DE DEVOPS: PRÁTICAS, DESAFIOS E RECOMENDAÇÕES PARA A
MELHORIA NA IMPLEMENTAÇÃO / Rayan Freitas. – 2025.

47 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus
de Russas, Curso de Ciência da Computação, Russas, 2025.

Orientação: Profa. Ma. Prof. Ms. Valéria Maria da Silva Pinheiro..

1. DevOps;. 2. automação;. 3. comunicação eficaz;. 4. segurança do software;. 5.
eficiência do desenvolvimento. I. Título.

CDD 005

RAYAN VICTOR DE SOUZA FREITAS

ADOÇÃO DE DEVOPS: PRÁTICAS, DESAFIOS E RECOMENDAÇÕES PARA A
MELHORIA NA IMPLEMENTAÇÃO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de Bacharel em Ciência da Computação.

Aprovada em: 13/03/2025.

BANCA EXAMINADORA

Prof. Ms. Valéria Maria da Silva
Pinheiro (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Ms. Hugo Nathan Barbosa Regis
Universidade Federal do Ceará - (UFC)

Prof. Ms. Fernanda Ferreira do Nascimento
Universidade Federal do Ceará - (UFC)

Dedico este trabalho a Deus, fonte de minha força e inspiração. Sua presença constante me guiou, me fortaleceu e me capacitou a superar os desafios dessa jornada. Que a Sua sabedoria continue a iluminar meu caminho e a me conduzir em tudo o que faço.

AGRADECIMENTOS

Primeiramente, **a Deus**, por me conceder discernimento, sabedoria e inteligência ao longo desses cinco anos de graduação. Sou profundamente grato por Sua orientação nos momentos difíceis e por transformar minha jornada universitária em uma oportunidade para viver Seus planos.

A minha mãe, minha namorada, familiares e amigos, que sempre buscaram tornar essa caminhada mais leve, me ensinando valores educacionais, profissionais e, acima de tudo, o amor ao próximo. Vocês são o meu espelho de vida. Agradeço também a toda minha família pelo apoio incondicional e pelo esforço dedicado à realização desta importante etapa.

A **Universidade Federal do Ceará**, que proporcionou meu crescimento pessoal e profissional por meio do ensino e dos projetos acadêmicos, aos profissionais que fazem parte do campus Russas, e, em especial, aos professores: **Profa. MS. Valéria Maria da Silva Pinheiro**, pela orientação excepcional; **Prof. MS. Hugo Nathan Barbosa Regis** e **Profa. MS. Fernanda Ferreira do Nascimento**, por aceitarem participar da banca avaliadora deste momento crucial da minha formação.

Por fim, agradeço à cidade de **Russas-CE** pelo acolhimento, pelas inúmeras experiências, amizades e parcerias que enriqueceram essa fase tão significativa da minha vida.

RESUMO

No Desenvolvimento de Software enfrentar desafios vem se tornando uma realidade para as equipes de desenvolvimento, desde de um planejamento mal elaborado até uma manutenção mal executada, os desenvolvedores se deparam com uma variedade de obstáculos que podem impactar o sucesso de um projeto. Estudos evidenciam que uma das maiores causas de desafios no ciclo de desenvolvimento de software reside na comunicação eficaz com diversas partes interessadas, incluindo clientes, gerentes de projeto e membros da equipe. A dificuldade em colaborar de forma eficiente com colegas de equipe em diferentes áreas pode levar a mal-entendidos, atrasos e até mesmo a falhas no projeto. Nesse contexto, o presente trabalho tem como objetivo investigar como a prática e cultura DevOps podem ser integradas de maneira eficaz no ciclo de vida do desenvolvimento de software, visando melhorar a comunicação entre equipes, automatizar processos e aumentar a eficiência na entrega de software. Para isso, serão implementadas práticas como integração contínua, entrega contínua e automação de testes, focando em um ambiente controlado de desenvolvimento onde se espera reduzir os atrasos e erros comuns no processo. Dessa forma, o trabalho busca fornecer evidências concretas de como o uso dessas práticas impacta a comunicação e eficiência em projetos de software. Ao adotar abordagens de DevOps, que enfatizam a colaboração entre equipes, a automação de tarefas e a integração contínua, espera-se enfrentar os desafios de comunicação, segurança e eficiência mencionados anteriormente, contribuindo para um desenvolvimento de software mais ágil, confiável e seguro.

Palavras-chave: : DevOps; comunicação eficaz; segurança do software; automação; eficiência do desenvolvimento

ABSTRACT

In Software Development, facing challenges has become a reality for development teams, from poorly prepared planning to poorly executed maintenance, developers are faced with a variety of obstacles that can impact the success of a project. Studies show that one of the biggest causes of challenges in the software development cycle lies in effective communication with various stakeholders, including customers, project managers and team members. Difficulty collaborating efficiently with teammates in different areas can lead to misunderstandings, delays, and even project failures. In this context, the present work aims to maximize the efficiency and quality of software development through the integration of DevOps practice and culture into the development life cycle. The aim is to improve and accelerate software delivery, ensure its security and promote process automation throughout all phases of the software life cycle. By adopting DevOps approaches, which emphasize collaboration between teams, task automation and continuous integration, it is expected to address the communication, security and efficiency challenges mentioned above, contributing to more agile, reliable and secure software development.

Keywords: : DevOps; effective communication; software security; automation; development efficiency

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas da Entrega Contínua	19
Figura 2 – Etapas da Integração Contínua	19
Figura 3 – Etapas da Implementação Contínua	20
Figura 4 – Etapas do Monitoramento Contínuo	21
Figura 5 – Etapas da Metodologia	26
Figura 6 – Questionário aceitação	29
Figura 7 – Questionário Curso	30
Figura 8 – Questionário projetos	31
Figura 9 – Questionário faixa etária	32
Figura 10 – Questionário Tamanho da Equipe	33
Figura 11 – Questionário papel na equipe	34
Figura 12 – Questionário nível de experiência	35
Figura 13 – Questionário familiaridade com o conceito DevOps	36
Figura 14 – Questionário frequência as equipes realizam deploy	37
Figura 15 – Questionário sobre quais ferramentas de DevOps os participantes conhecem ou já utilizaram	38
Figura 16 – Questionário sobre quais ferramentas de containerização os participantes conhecem ou já ouviram falar	39
Figura 17 – Questionário sobre quais ferramentas de monitoramento e logs os participantes conhecem ou já ouviram falar	40
Figura 18 – Questionário percepção dos participantes respeito dos principais objetivos do DevOps	41
Figura 19 – Questionário sobre as principais areas que necessitam de melhoria para facilitar a adoção das práticas DevOps	42

LISTA DE QUADROS

Quadro 1 – Comparação entre Trabalhos Relacionados	24
--	----

LISTA DE ABREVIATURAS E SIGLAS

CD	Contínuos Delivery
CI	Contínuos Integration
DevOps	Desenvolvimento e Operação
IC	Implementação Continua
ICs	Identificação dos Itens de Configuração
MC	Monitoramento Contínuo
TI	Tecnologia da Informação

SUMÁRIO

	LISTA DE QUADROS	8
1	INTRODUÇÃO	12
2	OBJETIVOS	14
2.1	Objetivo Geral	14
2.2	Objetivos Específicos	14
3	FUNDAMENTAÇÃO TEÓRICA	15
3.1	Gerência de Configuração	15
3.2	DevOps	17
3.2.1	<i>Práticas DevOps</i>	18
3.2.2	<i>Entrega Contínua (Contínuos Delivery)</i>	18
3.2.3	<i>Integração Contínua (Contínuos Integrations)</i>	18
3.2.4	<i>Integração Contínua (Continuous Integration)</i>	19
3.2.5	<i>Implementação Contínua (Contínuos Deployment)</i>	20
3.2.6	<i>Monitoramento Contínuo</i>	20
4	TRABALHOS RELACIONADOS	23
5	METODOLOGIA	26
5.1	Revisão da Literatura	27
5.2	Planejamento	27
5.2.1	<i>Questões de Pesquisa</i>	27
5.3	Elaboração do Questionário de Investigação	27
5.4	Aplicação do Questionário	28
5.5	Análise e Validação dos Dados	28
5.6	Refinamento e Propostas de Recomendações	28
6	RESULTADOS	29
6.1	Aplicação do Questionário	29
6.2	Resultados sobre o perfil dos participantes e dos times de desenvolvimento	30
6.3	Nível de conhecimento dos desenvolvedores sobre práticas DevOps . . .	32
6.4	Práticas e ferramentas DevOps mais conhecidas e utilizadas	33
6.5	Benefícios percebidos da adoção de práticas DevOps nas equipes	35
6.6	Maiores lacunas de conhecimento relacionadas a DevOps	39

6.7	Dificuldades enfrentadas por desenvolvedores com pouca experiência na adoção de DevOps	41
7	CONCLUSÕES E TRABALHOS FUTUROS	44
7.1	Conclusão	44
7.2	Considerações Finais	44
7.3	Trabalhos Futuros	45
	REFERÊNCIAS	46

1 INTRODUÇÃO

O desenvolvimento de software é uma atividade central na era digital, permeando quase todos os setores da economia e da sociedade. No entanto, apesar de seu papel crítico, o processo de desenvolvimento de software é frequentemente marcado por desafios como a complexidade crescente dos sistemas, prazos apertados, mudanças constantes nos requisitos e a necessidade de garantir a qualidade do produto final. Esses desafios podem resultar em problemas como atrasos na entrega, falhas de integração, baixa qualidade do código e dificuldades na manutenção e evolução dos sistemas (Ahmad et al., 2021)

Para enfrentar os desafios enfrentados no desenvolvimento de software, a Engenharia de Software se estabeleceu como uma área abrangente que oferece métodos, processos e ferramentas para garantir que o desenvolvimento seja realizado de forma eficiente e controlada. Nesse contexto, a gerência de configuração desempenha um papel essencial, permitindo o controle e a coordenação das mudanças nos artefatos de software, assegurando a integridade e a consistência dos sistemas ao longo de todo o seu ciclo de vida. A gerência de configuração abrange atividades como controle de versão, gestão de mudanças, auditoria de configuração e gestão de liberações, práticas fundamentais para o sucesso de projetos de software, conforme destacado por (Silva et al., 2022) e (Pressman, 2011)

Além disso, nas últimas décadas, o movimento conhecido como Desenvolvimento e Operação (DevOps) tem revolucionado a maneira como o software é desenvolvido e entregue. O DevOps, que integra as palavras "desenvolvimento" (*Development*) e "operações" (*Operations*), busca uma maior colaboração entre essas duas áreas, promovendo a automação de processos e a implementação de práticas como *Contínuos Integration (CI)* e *Contínuos Delivery (CD)*. Ao reduzir os ciclos de desenvolvimento, melhorar a eficiência operacional e promover uma cultura de colaboração, o DevOps se tornou uma estratégia indispensável para organizações que buscam agilidade e qualidade em seus processos. De acordo com (Sousa, 2019) e (Kim et al., 2013), o DevOps também enfatiza a experimentação e o feedback contínuo, permitindo que as equipes melhorem de maneira iterativa.

Ferramentas como *GitLab CI/CD*¹, *Docker*², *Ansible*³, *Grafana*⁴ e *Prometheus*⁵

¹ <https://about.gitlab.com/stages-devops-lifecycle/continuous-integration/>

² <https://www.docker.com/>

³ <https://www.ansible.com/>

⁴ <https://grafana.com/>

⁵ <https://prometheus.io/>

são fundamentais nesse ecossistema, facilitando a integração, automação e monitoramento dos processos de desenvolvimento e operações. O uso dessas ferramentas é amplamente estudado por Freitas (2018), Kumpulainen (2023), Valente (2023), Nallamala (2019) e Grafana Labs (2023).

Este estudo visa investigar a atual adoção das práticas DevOps em pequenos projetos de software, com foco especial em desenvolvedores que possuem experiência limitada na área. A pesquisa busca avaliar, de forma detalhada, o grau de conhecimento desses profissionais sobre as práticas DevOps, identificar as ferramentas mais comumente utilizadas e os principais obstáculos que enfrentam durante a implementação dessas práticas. Com base nos resultados obtidos, serão apresentadas recomendações práticas que visam otimizar a adoção de DevOps, levando em consideração as particularidades e os desafios encontrados em contextos de desenvolvimento menores.

2 OBJETIVOS

2.1 Objetivo Geral

O objetivo deste trabalho é analisar o estado atual da adoção de práticas DevOps em projetos de extensão desenvolvidos no ambiente acadêmico, avaliando o nível de conhecimento dos desenvolvedores, as ferramentas e metodologias mais utilizadas, com o intuito de fornecer recomendações para otimizar a implementação dessas práticas e facilitar sua adoção.

2.2 Objetivos Específicos

- Avaliar o nível de conhecimento dos desenvolvedores sobre DevOps, por meio de questionários direcionados;
- Identificar as práticas e ferramentas DevOps mais utilizadas, utilizando levantamento de dados coletados em questionários e entrevistas;
- Analisar os benefícios percebidos com a adoção de DevOps, com base nas respostas dos desenvolvedores
- Investigar as dificuldades enfrentadas na implementação de práticas DevOps, através de questões abertas no questionário e entrevistas;
- Propor recomendações para melhorar a adoção de DevOps, a partir da análise dos dados coletados e revisão de literatura sobre o tema.

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo oferece uma revisão abrangente dos conceitos fundamentais que sustentam este trabalho, começando pela gerência de configuração, um processo essencial no desenvolvimento de software focada no controle e coordenação das mudanças nos artefatos do sistema ao longo do ciclo de vida do software. Em seguida, será discutido o movimento DevOps, uma abordagem colaborativa entre as equipes de desenvolvimento e operações.

Concluindo este capítulo, é importante ressaltar que a gerência de configuração não apenas garante a integridade e a consistência dos sistemas de software, mas também facilita a comunicação e a colaboração entre diferentes equipes. Essa interação é fundamental em um ambiente cada vez mais dinâmico, onde as mudanças são frequentes e as expectativas de entrega rápida aumentam.

Ademais, o movimento DevOps representa uma evolução significativa na forma como as organizações abordam o desenvolvimento e a operação de software. A integração das equipes de desenvolvimento e operações promove uma cultura de colaboração, reduzindo silos e aumentando a eficiência. Este capítulo serve como um alicerce para entender como essas práticas não só são relevantes, mas também essenciais para o sucesso de pequenos projetos de software. Portanto, a compreensão desses conceitos permitirá a análise crítica e a aplicação prática das metodologias DevOps, contribuindo para a melhoria contínua dos processos de desenvolvimento em organizações de diversos tamanhos.

3.1 Gerência de Configuração

A gerência de configuração é um processo fundamental no desenvolvimento de software, concentrando-se no controle e gerenciamento das mudanças nos artefatos e componentes dos sistemas ao longo de seu ciclo de vida. É uma prática essencial para garantir a integridade, consistência e rastreabilidade do software em ambientes complexos de desenvolvimento.

De acordo com (Silva et al., 2022), a gerência de configuração é definida como "o processo de identificação, organização e controle dos itens de configuração; estabelecendo sua integridade, status de vida e relacionamento com outros itens, enquanto aborda as mudanças propostas e implementadas em cada item durante o ciclo de vida do sistema". O autor ressalta a importância da gerência de configuração no desenvolvimento de software, destacando sua capacidade de garantir a consistência e rastreabilidade dos elementos do sistema, bem como sua

vitalidade para a identificação e resolução de problemas.

(Pressman, 2011) afirma que a gerência de configuração desempenha um papel crucial ao longo de todo o ciclo de desenvolvimento de software, desde a solicitação de uma alteração até a auditoria dos itens entregues no final do desenvolvimento. Nesse contexto, ele identifica cinco atividades fundamentais que compõem a gerência de configuração de software:

Identificação dos Itens de Configuração (ICs): Esta atividade envolve a identificação e documentação dos diversos artefatos e componentes que compõem o sistema de software. Os ICs podem incluir código-fonte, documentos de design, arquivos de configuração, scripts de automação e outros artefatos relevantes. A correta identificação dos ICs é essencial para estabelecer uma base sólida para o controle e gerenciamento das mudanças ao longo do ciclo de vida do sistema.

Controle de Versão: O controle de versão é responsável por rastrear e gerenciar as alterações nos ICs ao longo do tempo. Isso é geralmente feito por meio de sistemas de controle de versão, como Git, Subversion () ou Mercurial. O controle de versão permite que os desenvolvedores trabalhem de forma colaborativa, acompanhem as mudanças feitas em cada versão do código e revertam para versões anteriores, se necessário.

Gestão de Mudanças: Esta atividade envolve o gerenciamento das solicitações de mudanças nos ICs, desde a sua proposição até a sua implementação e validação. Isso inclui o registro e documentação das solicitações de mudanças, avaliação de impacto, aprovação de mudanças e comunicação de alterações aos stakeholders (partes interessadas) relevantes. A gestão de mudanças visa garantir que todas as alterações sejam controladas e implementadas de forma eficaz, minimizando o impacto negativo no sistema.

Auditoria de Configuração: A auditoria de configuração é uma atividade de garantia de qualidade que visa verificar a conformidade dos ICs em relação às políticas e padrões estabelecidos pela organização. Isso pode incluir a verificação da integridade e consistência dos artefatos, a conformidade com as diretrizes de codificação, a presença de documentação adequada e o cumprimento de requisitos regulamentares. As auditorias de configuração ajudam a garantir que o sistema de software esteja em conformidade com os padrões e requisitos aplicáveis.

Gestão de Liberações: A gestão de liberações envolve o planejamento, coordenação e entrega das versões do sistema de software aos usuários finais. Isso inclui a preparação de pacotes de liberação, a realização de testes finais, a documentação de liberações e a comunicação de alterações aos usuários e stakeholders (partes interessadas). A gestão de liberações visa

garantir que as versões do software sejam entregues de forma consistente, segura e eficiente, atendendo às expectativas dos usuários e às necessidades do negócio.

Essas atividades são essenciais para garantir que o software mantenha sua integridade, rastreabilidade e consistência ao longo do processo de desenvolvimento, fornecendo uma base sólida para o gerenciamento eficaz das mudanças no software, segundo (Pressman, 2011).

3.2 DevOps

Nos últimos anos o termo "DevOps" tem ganhado destaque como uma abordagem inovadora no desenvolvimento de software e nas operações de Tecnologia da Informação (TI). DevOps, uma contração das palavras "Desenvolvimento" (Development) e "Operações" (Operations), é uma cultura, filosofia e conjunto de práticas que visa promover a colaboração e integração entre equipes de desenvolvimento de software e operações de TI (Sousa, 2019). De acordo com (Sousa, 2019) a definição mais consentânea encontrada na literatura é a de que DevOps é uma nova abordagem de entrega de aplicações informáticas (software), através da colaboração entre as equipes de desenvolvimento e operações, em oposição à abordagem tradicional em que o desenvolvimento e as operações estão separados nos seus silos organizacionais

O conceito de DevOps teve sua origem em 2008, quando Patrick Debois publicou o artigo *"Agile and Operations Infrastructure: How Infra-gile Are You?"* (Debois, 2008). Neste trabalho, Debois argumentou que a infraestrutura poderia se adaptar de forma ágil às mudanças constantes nos negócios, assim como as metodologias ágeis de desenvolvimento se ajustavam às mudanças do mercado. Em 2009, durante a conferência Velocity da O'Reilly, John Allspaw e Paul Hammond apresentaram *"10+ Deploys Per Day: Dev and Ops Cooperation at Flickr"* (Allspaw; Hammond et al., 2009), marcando uma ênfase na necessidade de integração plena entre as equipes de desenvolvimento e operações para impulsionar a agilidade na geração de novos produtos e serviços de TI. Como resultado desses eventos, no mesmo ano, ocorreu o primeiro DevOpsDay, uma série mundial de conferências técnicas que abordam temas relacionados ao desenvolvimento de software, operações de infraestrutura em TI e assuntos correlatos, realizado em Ghent, Bélgica.

3.2.1 Práticas DevOps

As práticas DevOps representam um conjunto de processos, ferramentas e filosofias que visam integrar as equipes de desenvolvimento e operações, promovendo maior colaboração, automação e eficiência no ciclo de vida do software. Essas práticas são essenciais para melhorar a agilidade, qualidade e escalabilidade das aplicações. Nas próximas seções, serão detalhadas as principais práticas de DevOps, incluindo CI, CD, automação de infraestrutura, monitoramento e colaboração entre equipes. Cada uma dessas práticas será explorada com maior profundidade, destacando seus benefícios e desafios na implementação, especialmente em pequenos projetos de software.

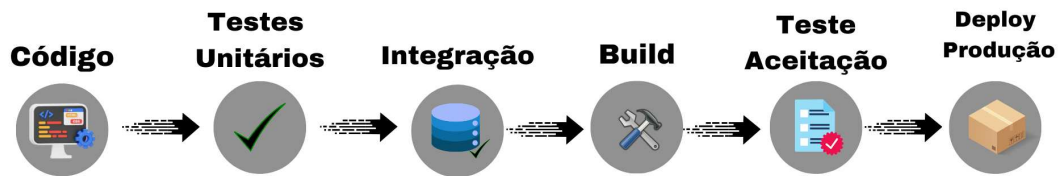
3.2.2 Entrega Contínua (Continuous Delivery)

O CD é uma prática fundamental no contexto do DevOps, envolvendo a integração frequente do código desenvolvido e a execução de testes automatizados. Essa abordagem visa garantir que as mudanças no código sejam integradas de forma contínua e que o software resultante seja testado de maneira eficiente (Rato; Silva, 2017). A CD possibilita uma abordagem ágil de desenvolvimento, com múltiplas integrações ao longo do dia, permitindo que o software seja entregue com maior rapidez e confiabilidade (Rato; Silva, 2017). A Figura 1 ilustra as etapas envolvidas na Integração Contínua. Esta figura mostra o fluxo de trabalho típico em um processo de CD, destacando como as mudanças no código são integradas, testadas e, eventualmente, implantadas.

3.2.3 Integração Contínua (Continuous Integrations)

A CI é uma prática que visa liberar novas versões de software de forma rápida e eficiente, possibilitando que as funcionalidades desenvolvidas estejam disponíveis para os usuários o mais rápido possível (Sharma; Coyne, 2015). A CI é essencial para o movimento DevOps, pois automatiza os processos de implantação de software, permitindo uma entrega automatizada e confiável (Levita; Mattar Mattar Neto, 2018). A Figura 2 ilustra as etapas envolvidas na CI. Esta figura apresenta o fluxo de trabalho típico em um processo de CI, destacando as diferentes fases, desde o desenvolvimento do código até a integração e a implantação. A automação desses processos permite que as mudanças no código sejam rapidamente integradas, testadas e disponibilizadas para os usuários, garantindo maior agilidade e qualidade no desenvolvimento.

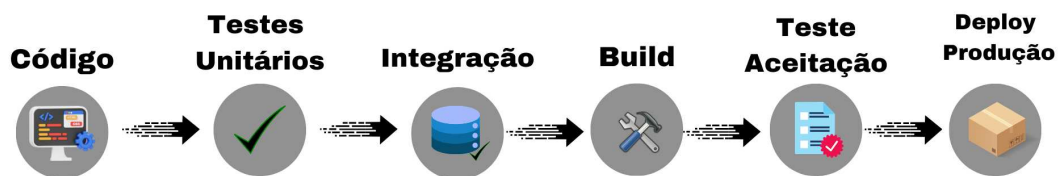
Figura 1 – Etapas da Entrega Contínua



Fonte: Elaborado pelo autor (2024).

3.2.4 Integração Contínua (Continuous Integration)

Figura 2 – Etapas da Integração Contínua

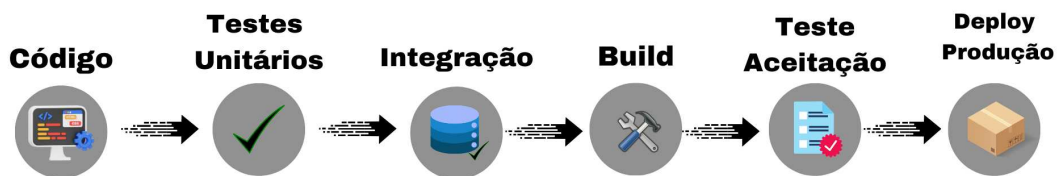


Fonte: Elaborado pelo autor (2024).

3.2.5 Implementação Contínua (*Continuous Deployment*)

O Implementação Contínua (IC) é uma prática que automatiza o processo de implantação de software, permitindo que as alterações desenvolvidas sejam automaticamente enviadas ao ambiente de produção após passarem pelos testes automatizados (Humble; Farley, 2015). Essa prática, repetida ao longo do ciclo de vida do software, garante uma entrega contínua de valor ao cliente, acelerando o ciclo de desenvolvimento e aumentando a eficiência operacional. A Figura 3 ilustra as etapas envolvidas na Implementação Contínua. Esta figura mostra o fluxo de trabalho típico em um processo de IC, desde o desenvolvimento e teste das alterações até a sua implantação automática no ambiente de produção. A automação desses processos assegura que as atualizações de software sejam entregues de forma rápida e confiável, permitindo uma resposta ágil às necessidades do cliente e melhorando a eficiência operacional.

Figura 3 – Etapas da Implementação Contínua



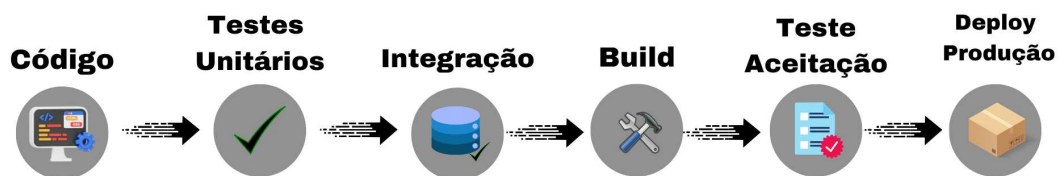
Fonte: Elaborado pelo autor (2024).

3.2.6 Monitoramento Contínuo

O Monitoramento Contínuo (MC) é uma prática essencial para acompanhar a evolução de um sistema ao longo do tempo, fornecendo métricas de desempenho e desenvolvimento que permitem uma visão clara do progresso do projeto (Levita; Mattar Neto, 2018). Essa prática é crucial para identificar e resolver problemas em tempo real, garantindo a qualidade opera-

cional do software desenvolvido. A Figura 4 ilustra as etapas envolvidas no MC. Esta figura demonstra como o monitoramento é realizado de forma contínua, desde a coleta de métricas e dados de desempenho até a análise e resposta a incidentes. O monitoramento contínuo permite uma visão abrangente da operação do sistema e ajuda a manter a estabilidade e a eficiência do software ao longo de seu ciclo de vida. Segundo (Kim et al., 2013), foram identificados três

Figura 4 – Etapas do Monitoramento Contínuo



Fonte: Elaborado pelo autor (2024).

princípios fundamentais para a adoção bem-sucedida do DevOps dentro de uma metodologia de desenvolvimento de software. O primeiro princípio busca estabelecer um modelo de trabalho otimizado que promova uma integração eficaz entre as equipes de desenvolvimento e operações. Para alcançar esse objetivo, propõe-se a decomposição dos processos em unidades menores e mais gerenciáveis. Essa abordagem incremental permite a otimização gradual de todo o fluxo de trabalho, enquanto se mantém um foco constante no controle e segurança das mudanças. Ao quebrar os processos em partes menores, a equipe pode identificar e corrigir problemas de forma mais ágil, mitigando riscos e promovendo uma abordagem iterativa para o desenvolvimento e entrega de software.

O segundo princípio promove a adoção de uma cultura de feedback contínuo, baseada em loops iterativos. Esses loops de feedback são projetados para identificar e corrigir proativamente as fontes de problemas no desenvolvimento, evitando retrabalho e melhorando a qualidade desde a origem do processo. Ao estabelecer canais de comunicação eficazes entre as

equipes de desenvolvimento e operações, esses loops garantem que feedbacks construtivos sejam fornecidos regularmente ao longo do ciclo de vida do projeto. Isso não apenas ajuda a resolver problemas de forma rápida e eficiente, mas também promove uma cultura de aprendizado e melhoria contínua, impulsionando a excelência operacional e a entrega de software de alta qualidade.

O terceiro princípio enfatiza a criação de uma cultura organizacional que valoriza a experimentação e encoraja uma mentalidade de aprendizado contínuo. Essa cultura reconhece a importância da repetição de processos como um meio essencial para adquirir domínio sobre as práticas de desenvolvimento e operações. Em vez de priorizar uma análise excessivamente detalhada, essa abordagem promove a ação e a aprendizagem prática como meio de progresso. Ao encorajar a experimentação e a disposição para tentar novas abordagens, as equipes podem descobrir novas soluções e identificar oportunidades de melhoria de forma mais eficaz. Isso não apenas promove a inovação e a adaptação às mudanças, mas também cria um ambiente que valoriza a aprendizagem e o crescimento contínuos.

4 TRABALHOS RELACIONADOS

Este capítulo aborda como os conceitos discutidos, como CI/CD e ferramentas, são aplicados em projetos para melhorar a eficiência e automação no desenvolvimento de software. Foi revisado como técnicas similares foram utilizadas em diferentes contextos, destacando melhores práticas e lições aprendidas para otimizar processos de desenvolvimento e entrega de software.

O trabalho de Pavan *et al.* (2018) apresenta uma análise detalhada sobre a implementação de práticas de DevOps no desenvolvimento e operações de software. O estudo investiga como o DevOps pode integrar equipes de desenvolvimento e operações para otimizar o processo de entrega de software e aumentar a colaboração entre as partes envolvidas. O foco principal é a adoção de práticas como CI e CD, e como essas práticas influenciam a eficiência e a qualidade do software. A pesquisa mostra que a adoção de práticas DevOps pode reduzir significativamente o tempo de ciclo de desenvolvimento e melhorar a qualidade do produto final, embora também enfrente desafios como mudanças culturais e adaptação a novas ferramentas e processos.

O estudo de Jones e Smith (2019) oferece uma análise abrangente sobre as práticas e princípios de DevOps no contexto do desenvolvimento e operações de software. Explora como a adoção de práticas DevOps pode transformar o desenvolvimento de software, destacando a importância da colaboração entre equipes, automação e a implementação de pipelines de CI/CD. Os resultados indicam uma melhoria significativa na eficiência dos processos de desenvolvimento e na qualidade dos produtos finais, com uma redução de 30% no tempo de entrega do software e uma melhoria de 90% na capacidade de resposta a incidentes.

O trabalho de Shah (2018) fornece uma revisão abrangente das práticas de CI/CD, destacando técnicas, ferramentas e melhores práticas para uma implementação eficaz dessas metodologias. O estudo revela que a adoção de CI/CD pode levar a uma melhoria substancial na eficiência e na qualidade do desenvolvimento de software, com uma redução de até 50% no tempo de desenvolvimento e uma melhoria de 60% na taxa de sucesso das entregas de software. A pesquisa enfatiza a importância da automação e integração contínua em todos os estágios do desenvolvimento.

A pesquisa de (Peters, 2020) examina o uso do GitLab para CI/CD e como ele facilita práticas DevOps. O estudo demonstra que o GitLab pode otimizar o desenvolvimento e a entrega de software, com uma redução de até 40% no tempo de entrega e um aumento de 50% na eficiência dos processos de desenvolvimento. A integração das práticas DevOps com o

GitLab promove uma colaboração mais eficaz e uma visibilidade melhorada em todo o ciclo de vida do software.

O trabalho de (Tan et al., 2021) explora como o Docker pode melhorar a eficiência e flexibilidade em ambientes de desenvolvimento e produção, focando na prática de containerização e orquestração. A pesquisa mostra que o Docker proporciona uma melhoria significativa na eficiência dos processos de desenvolvimento e na flexibilidade dos ambientes de produção, com uma redução de 50% nos problemas relacionados à configuração e compatibilidade de ambientes. A adoção do Docker resultou em uma melhoria de 60% na eficiência do processo de desenvolvimento e contribuiu para uma maior agilidade na implementação e gerenciamento de aplicações.

O Quadro 1 apresenta uma comparação dos principais trabalhos relacionados, destacando o foco principal, as práticas de DevOps adotadas e sua relevância para o presente projeto.

Quadro 1 – Comparação entre Trabalhos Relacionados

Trabalho	Foco Principal	Práticas de DevOps Adotadas	Relevância para o Projeto
Pavan <i>et al.</i> (2018)	Implementação de DevOps	CI/CD, automação de testes	Destaca a importância da integração entre equipes e práticas de CI/CD
Jones e Smith (2019)	Práticas e princípios de DevOps	CI/CD, monitoramento, colaboração	Demonstra impacto positivo de CI/CD e ferramentas de monitoramento
Shah (2018)	Revisão de CI/CD	Integração Contínua, Entrega Contínua	Fornecer uma visão geral das melhores práticas e benefícios de CI/CD
Peters (2020)	Uso do GitLab para CI/CD	GitLab CI/CD, automação de builds	Enfatiza o papel do GitLab na automação e colaboração
Tan (2021)	Uso do Docker para containerização	Containerização, orquestração com Docker	Ressalta a importância da containerização para a agilidade e consistência

Fontes: PAVAN et al. (2018). Detectando e propondo melhorias em cenário DevOps. Anais da ERES.
 JONES (2019). DevOps: Práticas e princípios. DevOps.com.
 SHAH (2018). CI/CD Pipeline Automation. ResearchGate.
 PETERS (2020). GitLab Commit 2020: Agile Compliance and Risk Operations. Techstrong TV.
 TAN (2021). DevOps: Multi-Stage Docker Container. Medium.

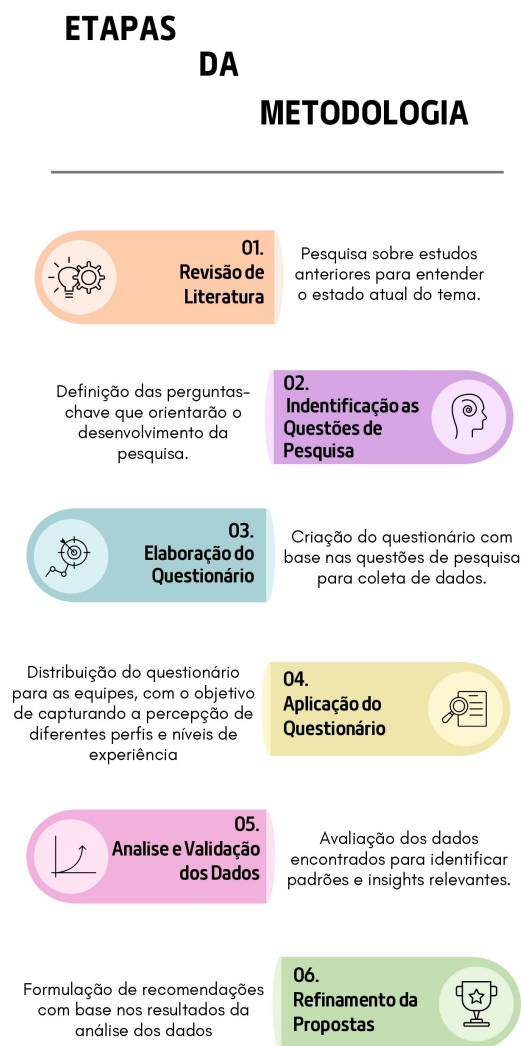
Apesar das contribuições importantes dos trabalhos mencionados, nenhum deles foca especificamente na adoção de práticas DevOps em **projetos de extensão no ambiente acadêmico**, que é o contexto principal do meu estudo. Enquanto os trabalhos revisados abordam cenários mais gerais ou corporativos, meu trabalho se concentra em avaliar como desenvolvedores em formação e inseridos em projetos de extensão universitários aplicam práticas DevOps, considerando suas particularidades e limitações.

Esse enfoque permitirá uma análise mais detalhada sobre as dificuldades e necessidades específicas desses desenvolvedores em um ambiente educacional, proporcionando recomendações adaptadas e estratégias que visem melhorar a adoção de DevOps nesse contexto específico. Além disso, o estudo trará uma contribuição prática para a comunidade acadêmica, incentivando uma formação mais voltada às práticas modernas do mercado.

5 METODOLOGIA

O objetivo desta pesquisa foi explorar a percepção dos desenvolvedores sobre a adoção de práticas DevOps. O estudo buscou entender o nível de familiaridade dos desenvolvedores com o conceito de DevOps, identificar as práticas e ferramentas mais conhecidas e utilizadas, além de investigar os benefícios percebidos ao adotar DevOps em suas equipes de desenvolvimento. A Figura 5 caracteriza as etapas metodológicas adotadas neste trabalho.

Figura 5 – Etapas da Metodologia



Fonte: Elaborado pelo autor (2024).

5.1 Revisão da Literatura

Nesta etapa, foi realizado um levantamento de trabalhos relevantes sobre práticas DevOps, automação eficaz, segurança do software e eficiência no desenvolvimento. Utilizou-se o Google Acadêmico como ferramenta principal de busca, aplicando filtros rigorosos com palavras-chave como "DevOps", "engenharia de software", "CI/CD", "Docker", "GitLab" e "implantação". Os trabalhos selecionados foram analisados e resumidos, destacando as principais ideias e contribuições, oferecendo uma base sólida para a investigação proposta.

5.2 Planejamento

A fase de planejamento inclui a identificação das **perguntas do questionário**, formulando perguntas-chave para compreender o impacto e a percepção das práticas DevOps entre desenvolvedores, especialmente aqueles com menos experiência em projetos pequenos no campus. As **perguntas do questionário** abordam conhecimento, uso de ferramentas, benefícios percebidos e as dificuldades enfrentadas pelos desenvolvedores.

5.2.1 Questões de Pesquisa

As principais questões formuladas para guiar esta pesquisa são:

1. Qual o nível de conhecimento dos desenvolvedores sobre práticas DevOps?
2. Quais práticas e ferramentas DevOps são mais conhecidas e utilizadas por esses desenvolvedores?
3. Quais são os benefícios percebidos da adoção de práticas DevOps nas equipes de desenvolvimento?
4. Quais são as principais dificuldades enfrentadas por desenvolvedores com pouca experiência na adoção de DevOps?
5. Em quais áreas os desenvolvedores demonstram maiores lacunas de conhecimento relacionadas a DevOps?

5.3 Elaboração do Questionário de Investigação

Um questionário abrangente foi desenvolvido para coletar informações dos desenvolvedores, com foco especialmente nos estudantes universitários envolvidos em projetos de

desenvolvimento realizados no campus da Universidade Federal do Ceará. O questionário possuía perguntas que avaliavam o conhecimento atual dos participantes, suas práticas DevOps e a percepção sobre os benefícios e desafios. Também foram incluídas questões de perfil, como idade, formação acadêmica e tempo de experiência. O questionário foi revisado para garantir clareza e relevância e ficou disponível no seguinte link: <https://forms.gle/ymGa1R9ke1oM981P7>.

5.4 Aplicação do Questionário

Após a elaboração, o questionário foi distribuído entre grupos de desenvolvedores selecionados, garantindo uma amostra representativa que abrange diferentes níveis de experiência. A coleta de dados foi realizada de forma sistemática, garantindo a qualidade e relevância das respostas. Os seguintes projetos foram incluídos na pesquisa: Nerds, BMO, Mais Informa UFC, LearningLab, LUDI, Fiserv, Include Jr e InsightLab. O questionário esteve disponível para coleta de respostas no período de 16 de dezembro de 2024 a 13 de janeiro de 2025. A distribuição foi realizada por meio dos e-mails dos professores orientadores dos projetos e por meio dos grupos de WhatsApp específicos de cada projeto.

5.5 Análise e Validação dos Dados

Com os dados coletados, foi realizada uma análise detalhada para identificação de padrões e insights. A análise focou no nível de conhecimento dos desenvolvedores sobre DevOps, nas dificuldades enfrentadas e nas áreas que necessitaram de suporte adicional. A validação dos dados garantida

5.6 Refinamento e Propostas de Recomendações

Com base na análise dos dados e no feedback, foram formuladas recomendações para melhorar a adoção de práticas DevOps nas equipes de desenvolvimento. As recomendações abordaram as lacunas de conhecimento identificadas, oferecendo sugestões práticas para superar as dificuldades e promover os benefícios do DevOps de forma eficaz

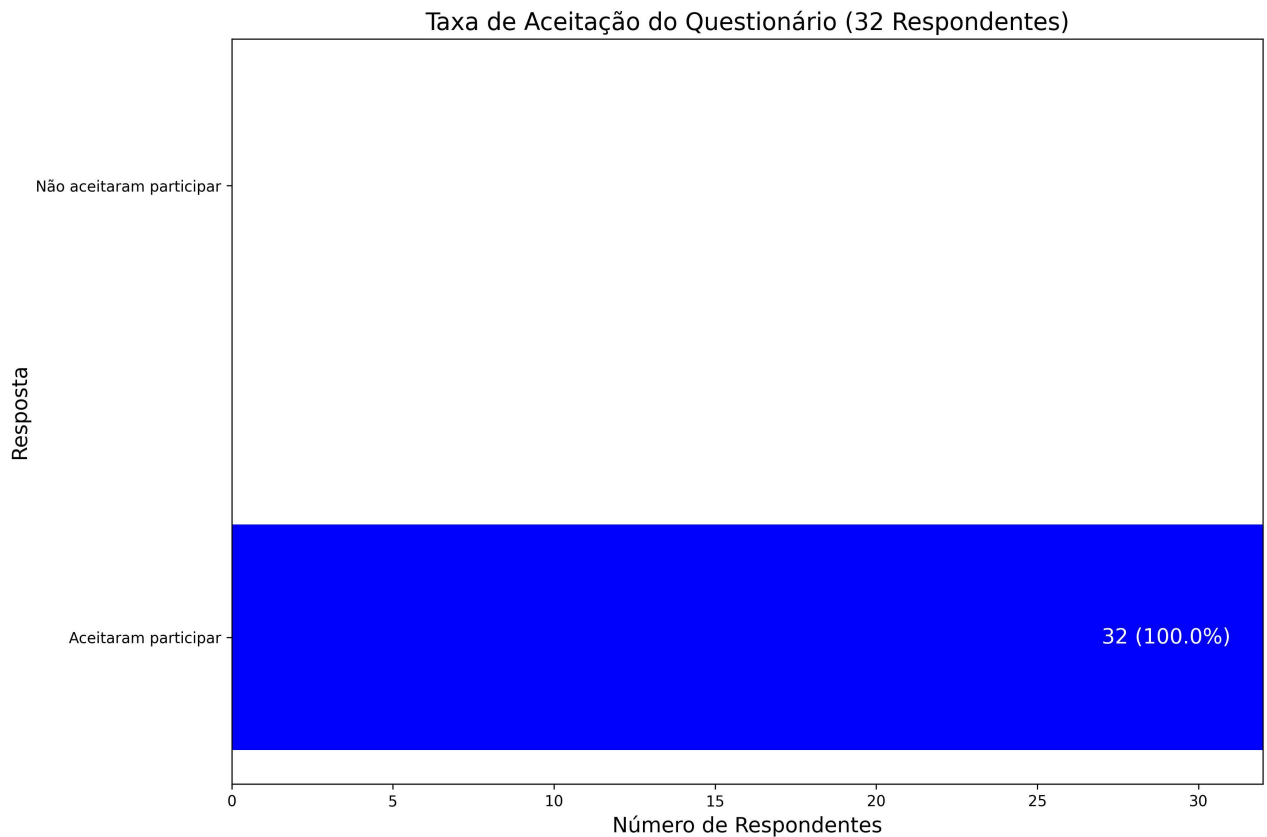
6 RESULTADOS

Neste capítulo, são apresentados os resultados alcançados ao longo do desenvolvimento desta pesquisa. A seguir, cada tópico aborda aspectos específicos relacionados aos objetivos do estudo.

6.1 Aplicação do Questionário

A aplicação do questionário teve como objetivo principal *analisar o estado atual da adoção de práticas DevOps* em projetos de extensão desenvolvidos no ambiente acadêmico. A pesquisa visou entender a percepção dos participantes sobre as práticas DevOps e investigar como essas práticas são implementadas nos projetos, buscando identificar desafios e oportunidades de melhoria. Antes da participação, os respondentes foram apresentados a um termo de consentimento, no qual estavam descritos os objetivos da pesquisa e a possibilidade de escolha em participar ou não. Na Figura 6, observa-se a aceitação dos participantes à pesquisa.

Figura 6 – Questionário aceitação



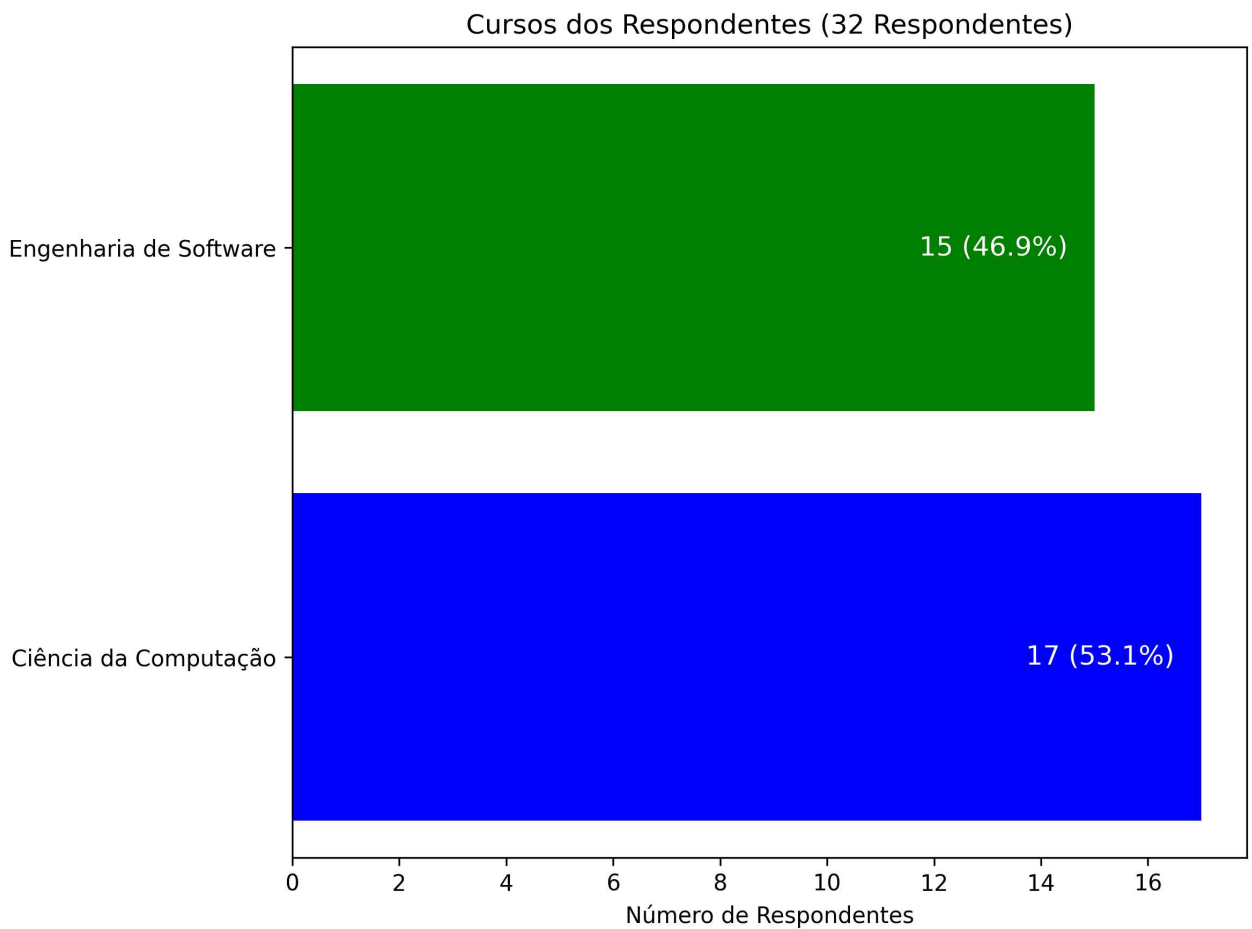
Fonte: Elaborado pelo autor (2024).

O questionário obteve um total de 32 respostas, com **100% dos participantes concordando** em participar da pesquisa, o que possibilitou a continuidade do preenchimento do formulário por todos os respondentes.

6.2 Resultados sobre o perfil dos participantes e dos times de desenvolvimento

Na Figura 7, é possível observar o curso dos participantes.

Figura 7 – Questionário Curso



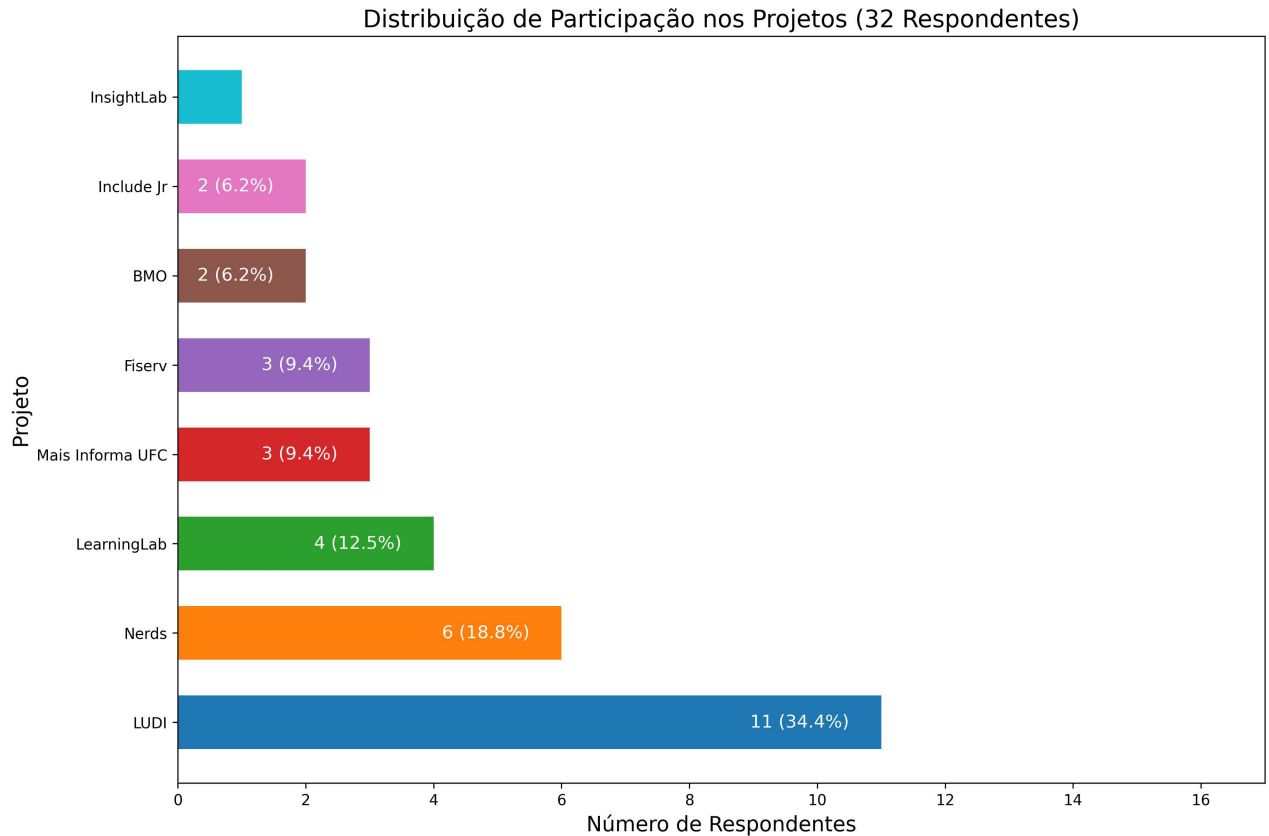
Fonte: Elaborado pelo autor (2025).

Em relação ao curso, os resultados mostraram que, das 32 respostas obtidas, 17 participantes (53,1%) pertencem ao curso de **Ciência da Computação**, enquanto os outros 15 participantes (46,9%) são do curso de **Engenharia de Software**.

A Figura 8 apresenta os resultados sobre os projetos nos quais os participantes estão atualmente envolvidos.

Em relação aos projetos, observou-se uma distribuição variada entre os participantes

Figura 8 – Questionário projetos



Fonte: Elaborado pelo autor (2025).

da pesquisa, com um total de 32 respostas. A participação nos projetos foi a seguinte: **11 pessoas** (34,4%) participam do projeto **LUDI**, **6 pessoas** (18,8%) estão envolvidas no projeto **Nerds**, **4 pessoas** (12,5%) estão no **LearningLab**, **3 pessoas** (9,4%) estão no **Mais Informa UFC**, **3 pessoas** (9,4%) estão no **Fiserv**, **2 pessoas** (6,3%) estão no **BMO**, **2 pessoas** (6,3%) estão no **Include Jr**, e **1 pessoa** (3,1%) participa do **InsightLab**.

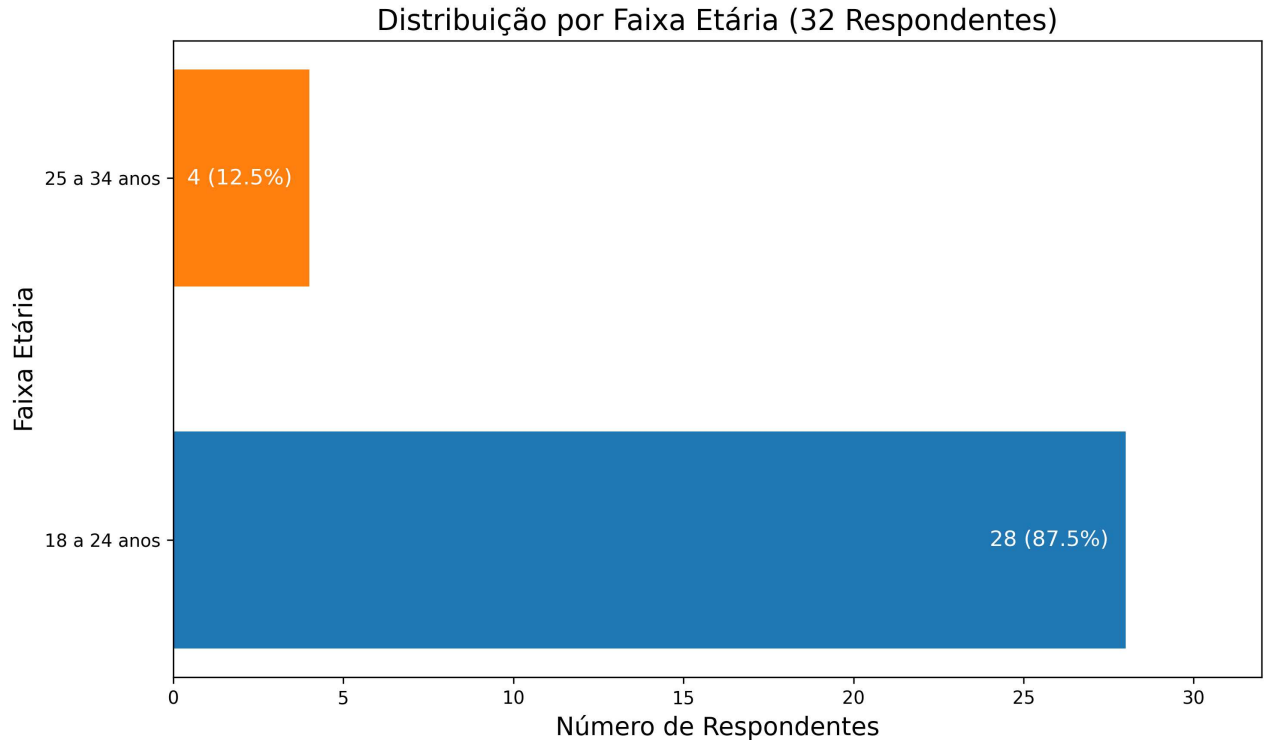
Na Figura 9, é possível ver as faixas etárias dos participantes.

Em relação à faixa etária, observou-se que dos 32 participantes, a maioria (87,5%) está na faixa etária de **18 a 24 anos**, enquanto 12,5% dos respondentes estão na faixa de **25 a 34 anos**.

Na Figura 10, é apresentado o gráfico que ilustra o tamanho das equipes de desenvolvimento dos participantes.

Em relação ao tamanho da equipe, observou-se que a maior parte das equipes é composta por **11 a 20 pessoas**, representando 14 participantes (43,6%). As equipes com **6 a 10 pessoas** e com **1 a 5 pessoas** apresentaram uma distribuição igual, com 9 participantes (28,1%) em cada categoria.

Figura 9 – Questionário faixa etária



Fonte: Elaborado pelo autor (2025).

Na Figura 11, apresenta o gráfico que ilustra os papéis dos participantes dentro das equipes de desenvolvimento.

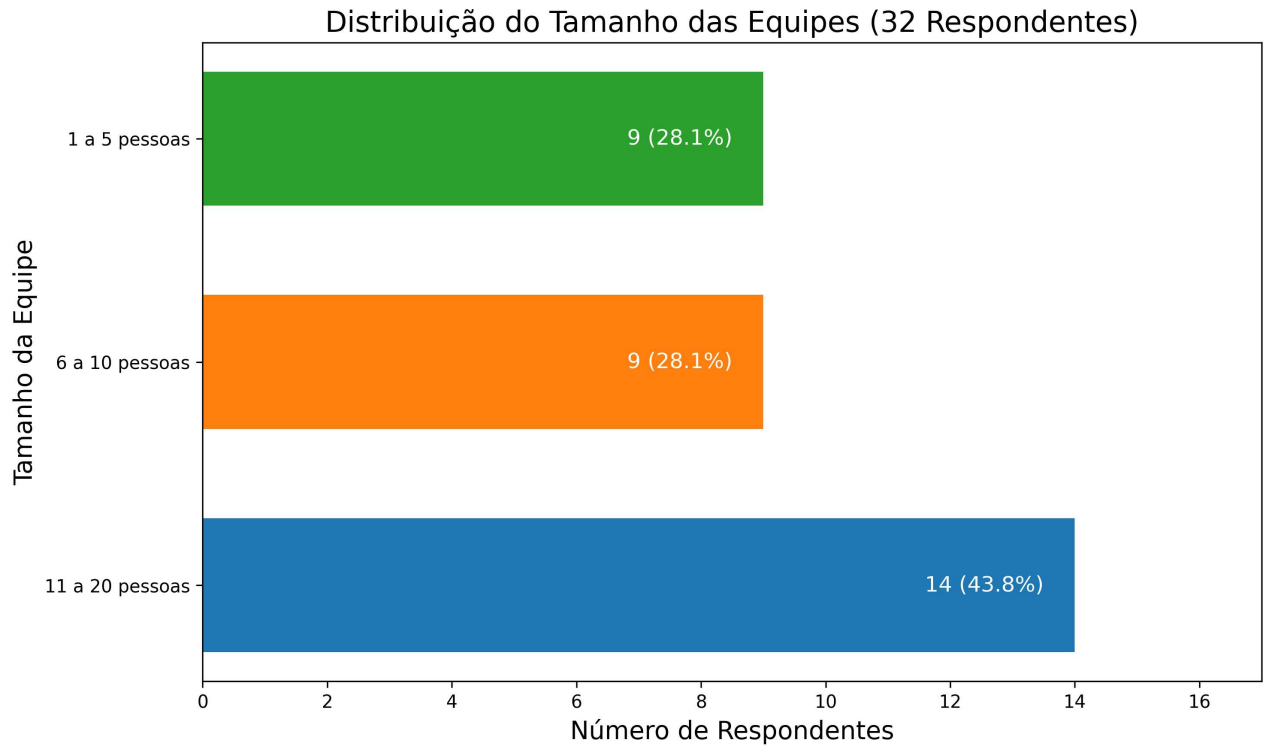
Em relação ao papel dentro da equipe, observou-se que a maioria dos participantes atua como **desenvolvedor Full Stack**, representando 10 participantes (31,3%). Os papéis de **desenvolvedor Front-end** e **outro** estão empatados, com 7 participantes (21,9%) cada. Em seguida, estão os **desenvolvedores Back-end**, com 6 participantes (18,8%), e, por último, os **analistas de sistemas**, com 2 participantes (6,3%). Ao todo, foram obtidas 32 respostas.

6.3 Nível de conhecimento dos desenvolvedores sobre práticas DevOps

Na Figura 12, é possível observar o nível de experiência em desenvolvimento de software dos participantes.

Em relação ao nível de experiência, observou-se que a maior parte dos participantes possui **de 1 a 3 anos de experiência**, totalizando 14 respondentes (43,8%). Em seguida, 11 participantes (34,4%) indicaram ter **de 3 a 5 anos de experiência**, 6 participantes (18,8%) possuem **menos de 1 ano de experiência**, e 1 participante (3,1%) relatou ter **de 5 a 10 anos de experiência**.

Figura 10 – Questionário Tamanho da Equipe



Fonte: Elaborado pelo autor (2025).

Na Figura 13, apresenta o gráfico que ilustra o nível de familiaridade dos participantes com o conceito de DevOps, evidenciando diferentes graus de conhecimento e experiência entre os respondentes.

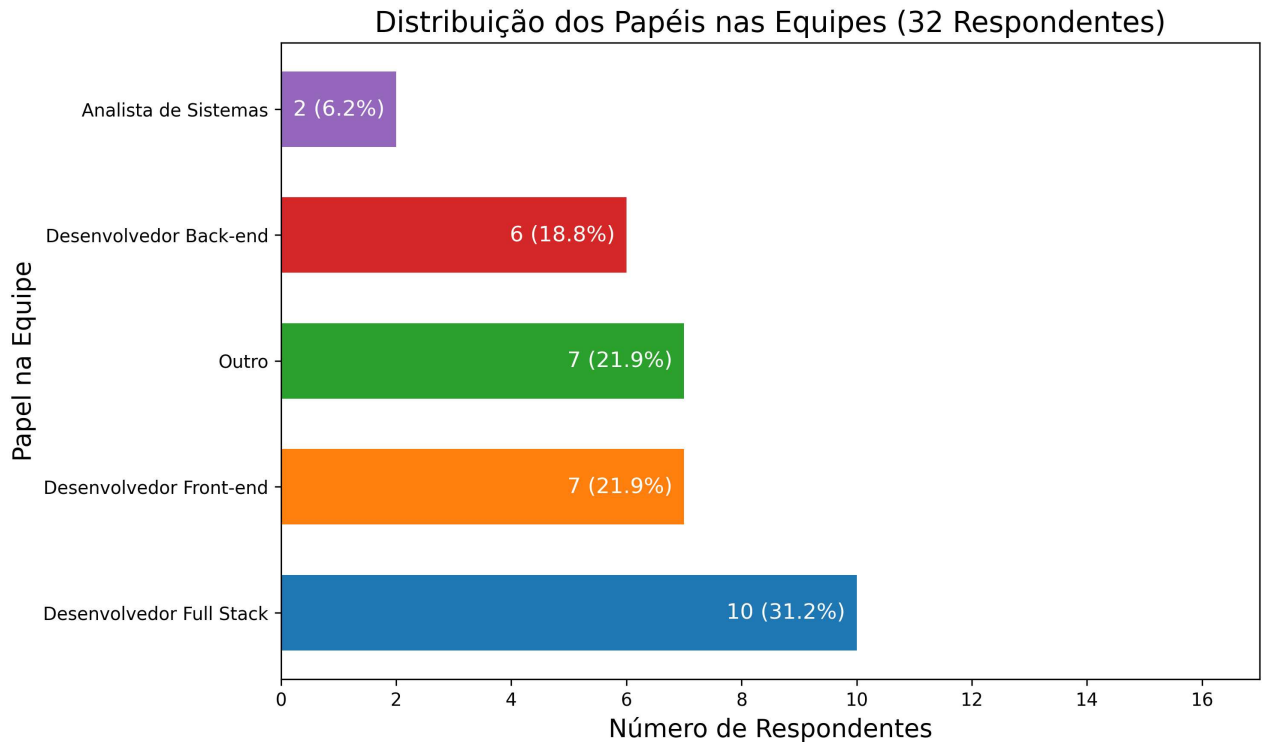
Em relação ao nível de familiaridade com o conceito de DevOps, observou-se que a metade dos participantes (16, correspondendo a 50%) **já ouviu falar do conceito, mas não possui experiência prática**. Além disso, os grupos de participantes que **nunca ouviram falar de DevOps** e os que são **muito familiarizados e utilizam regularmente o conceito** estão empatados, com 6 participantes cada (18,8%). Por fim, 4 participantes (12,5%) relataram ter **alguma experiência prática com DevOps**.

6.4 Práticas e ferramentas DevOps mais conhecidas e utilizadas

Na Figura 14, apresenta o gráfico que ilustra a frequência com que as equipes realizam deploys em seus projetos, proporcionando uma visão clara sobre os hábitos de entrega de software das equipes.

Em relação à frequência com que as equipes realizam deploys, observou-se que a maior parte realiza deploys **semanalmente**, representando 10 participantes (31,3%). Em seguida,

Figura 11 – Questionário papel na equipe



Fonte: Elaborado pelo autor (2025).

8 participantes (25%) indicaram que **nunca** realizam deploys, enquanto 6 participantes (18,8%) realizam deploys **diariamente**. Outros 5 participantes (15,6%) afirmaram realizar deploys **quinzenalmente**, e, por fim, 3 participantes (9,4%) realizam deploys **mensalmente**. Ao todo, foram obtidas 32 respostas.

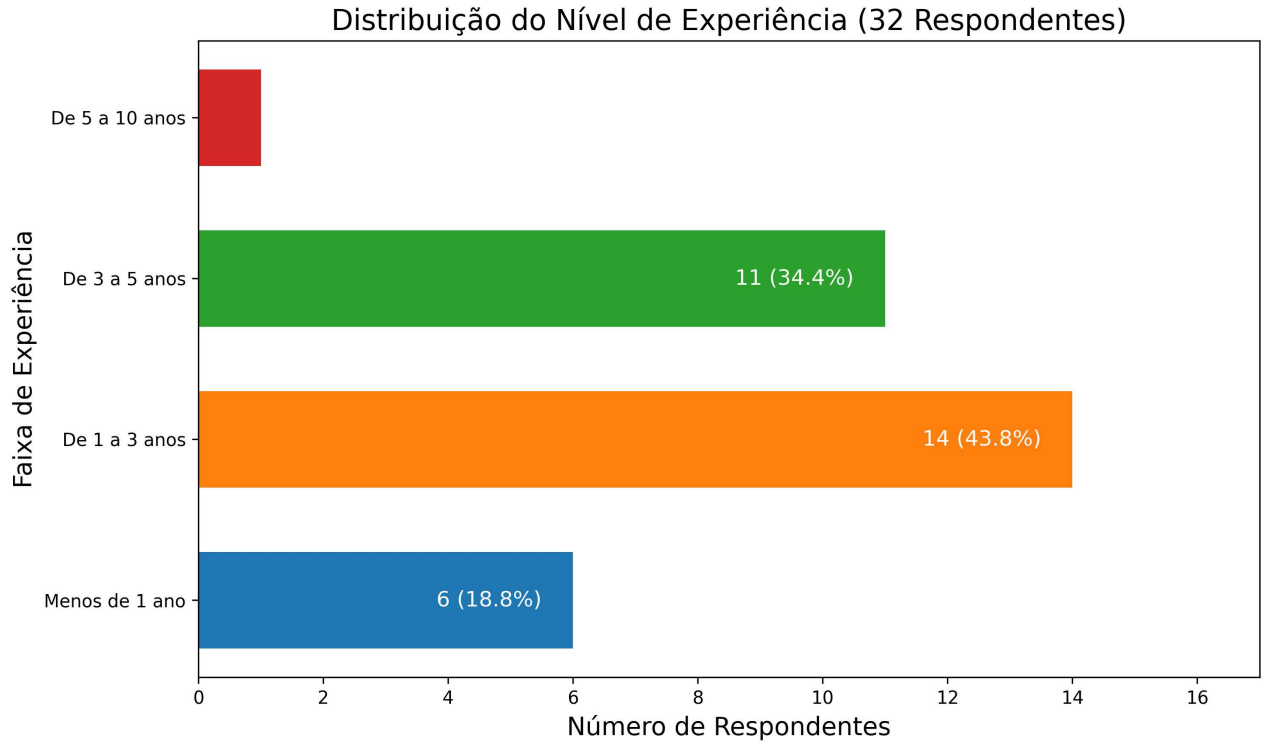
Na Figura 15, apresenta-se o gráfico do questionário sobre as ferramentas de DevOps que os participantes conhecem ou já utilizaram, oferecendo uma visão clara sobre a familiaridade dos respondentes com as soluções disponíveis no mercado.

Entre as ferramentas de DevOps conhecidas ou já utilizadas pelos participantes, o destaque foi para **Nenhum**, com 13 menções, correspondendo a (40,7%) do total. Em seguida, **GitLab CI/CD** foi apontado por 8 participantes (25,0%). Ferramentas como **Jenkins** foram citadas por 6 pessoas (18,8%). **Azure DevOps** apareceu em 2 respostas (6,3%), enquanto **Ansible**, **Terraform** e **GitHub Actions** foram mencionados 1 vez cada (3,1%).

Na Figura 16 apresenta-se o gráfico do questionário sobre as ferramentas de containerização que os participantes conhecem ou já ouviram falar, proporcionando uma visão sobre o nível de familiaridade dos respondentes com essas tecnologias.

Entre as ferramentas de containerização conhecidas ou das quais os participantes já

Figura 12 – Questionário nível de experiência



Fonte: Elaborado pelo autor (2025).

ouviram falar, o destaque foi para o **Docker**, com 18 respostas, representando 56,3% do total. Em seguida, **Kubernetes** foram mencionados por 7 participantes, correspondendo a 21,9%. O item **Nenhum** foi citado por 5 pessoas (15,6%) e, por fim, o **Podman** apareceu em 2 respostas (6,3%).

Na Figura 17, apresenta-se o gráfico do questionário sobre as ferramentas de monitoramento e logs que os participantes conhecem ou já ouviram falar, oferecendo uma visão geral sobre a familiaridade dos respondentes com essas tecnologias.

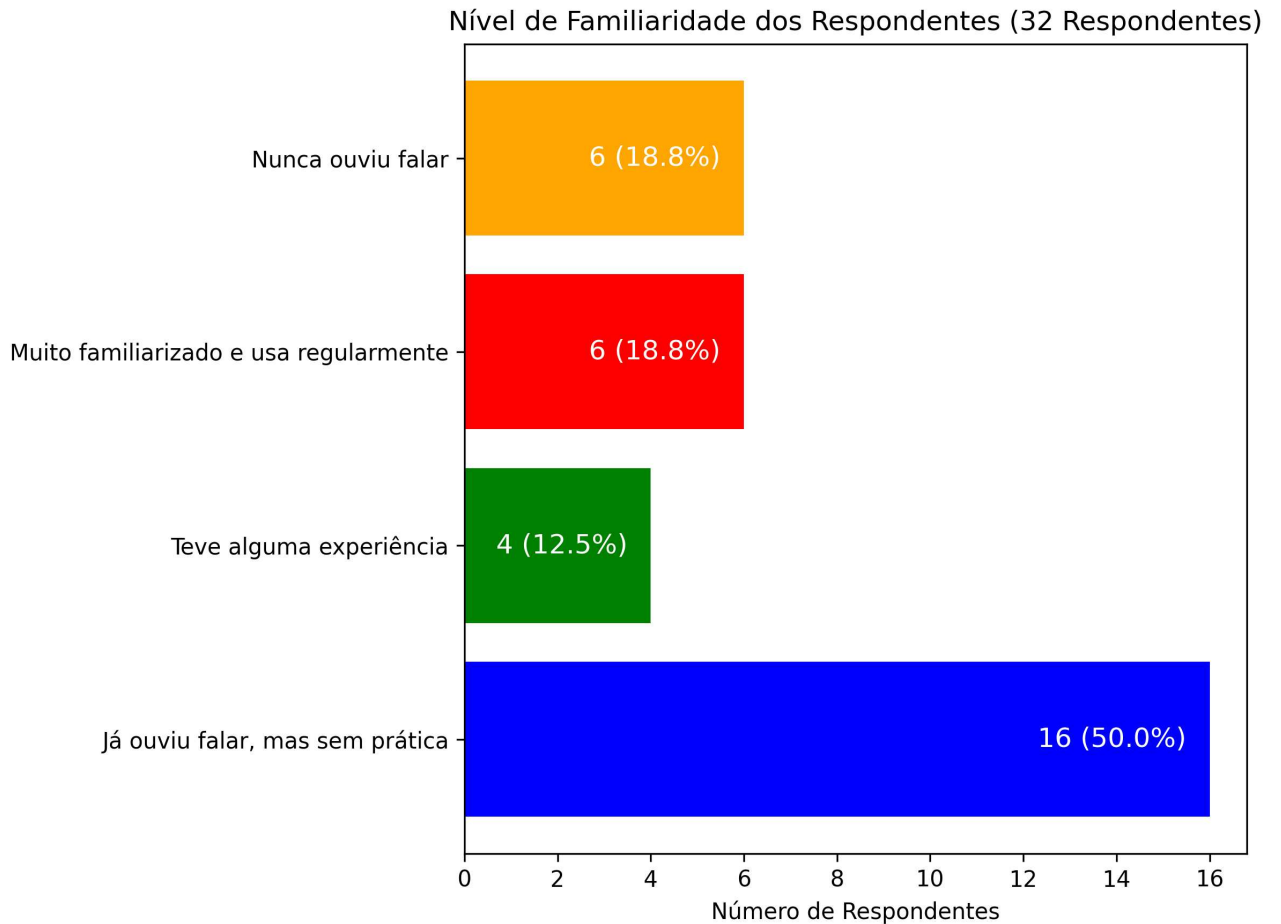
Entre as ferramentas de monitoramento e logs conhecidas ou já utilizadas pelos participantes, a opção **Nenhum** foi a mais mencionada, com 19 respostas. Em seguida, **Elasticsearch** foi citado por 5 participantes, enquanto **Prometheus** apareceu em 4 respostas. Tanto **Grafana** quanto **DataDog** foram mencionados por 2 participantes cada.

6.5 Benefícios percebidos da adoção de práticas DevOps nas equipes

Na Figura 18 apresenta-se a percepção dos participantes sobre os principais objetivos do DevOps, oferecendo uma visão clara das prioridades e metas identificadas pelas equipes.

Entre os principais objetivos do DevOps, na percepção dos participantes, o mais

Figura 13 – Questionário familiaridade com o conceito DevOps

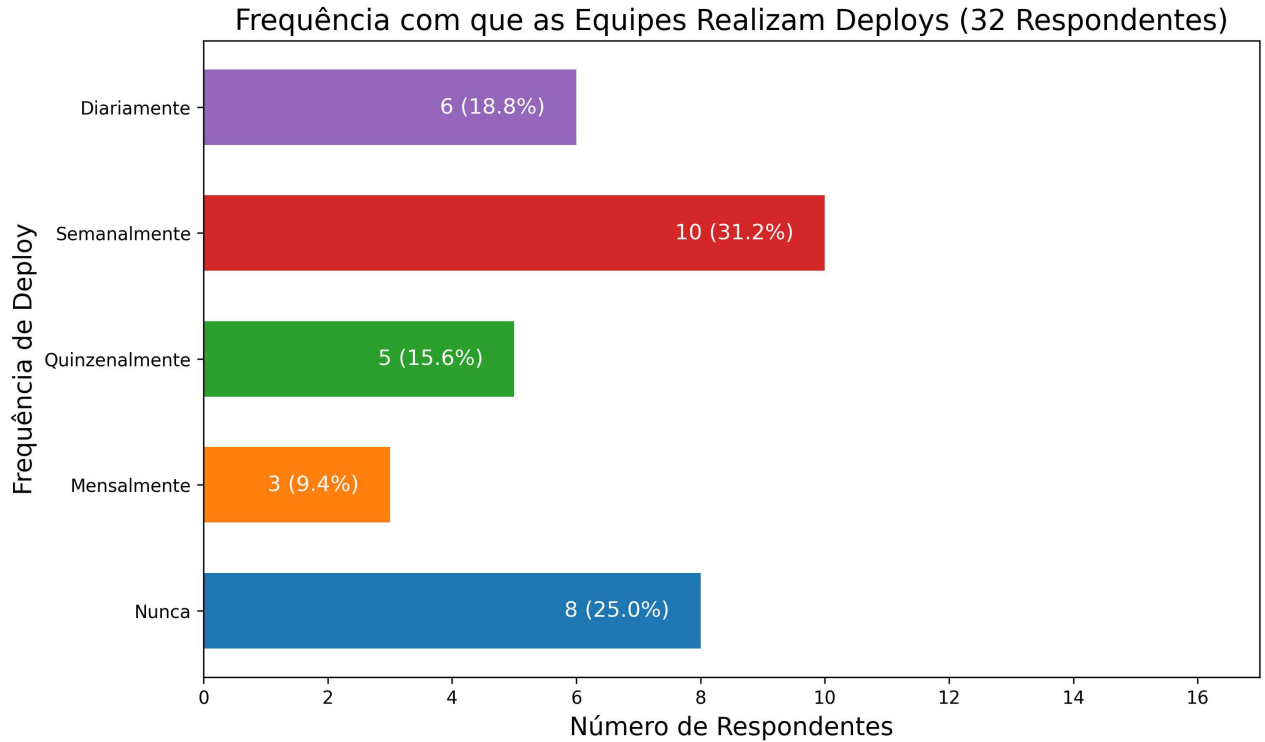


Fonte: Elaborado pelo autor (2025).

destacado foi **Aumentar a velocidade e a frequência dos deploys**, mencionado por 12 pessoas, o que representa 37,5% do total. Em seguida, **Automatizar processos para aumentar a eficiência e melhorar a qualidade da entrega do software** foi apontado por 9 participantes (28,1%). Já o objetivo **Promover a integração e colaboração entre equipes** recebeu 6 menções (18,8%), enquanto **Melhorar a qualidade da entrega do software** foi citado por 5 participantes (15,6%). No total, foram contabilizadas 32 respostas, evidenciando a importância da eficiência, da velocidade e da colaboração no contexto das práticas DevOps.

Os benefícios percebidos com a adoção das práticas DevOps indicam uma transformação significativa nos processos de desenvolvimento e operação de software. A **melhora na integração e automação do software**, citada com maior frequência, sugere que as equipes veem a automação como uma chave para aumentar a eficiência e a agilidade. Isso ocorre porque a automação de processos elimina tarefas manuais repetitivas, permitindo que os desenvolvedores e operadores se concentrem em atividades mais estratégicas, como a resolução de problemas

Figura 14 – Questionário frequência as equipes realizam deploy



Fonte: Elaborado pelo autor (2025).

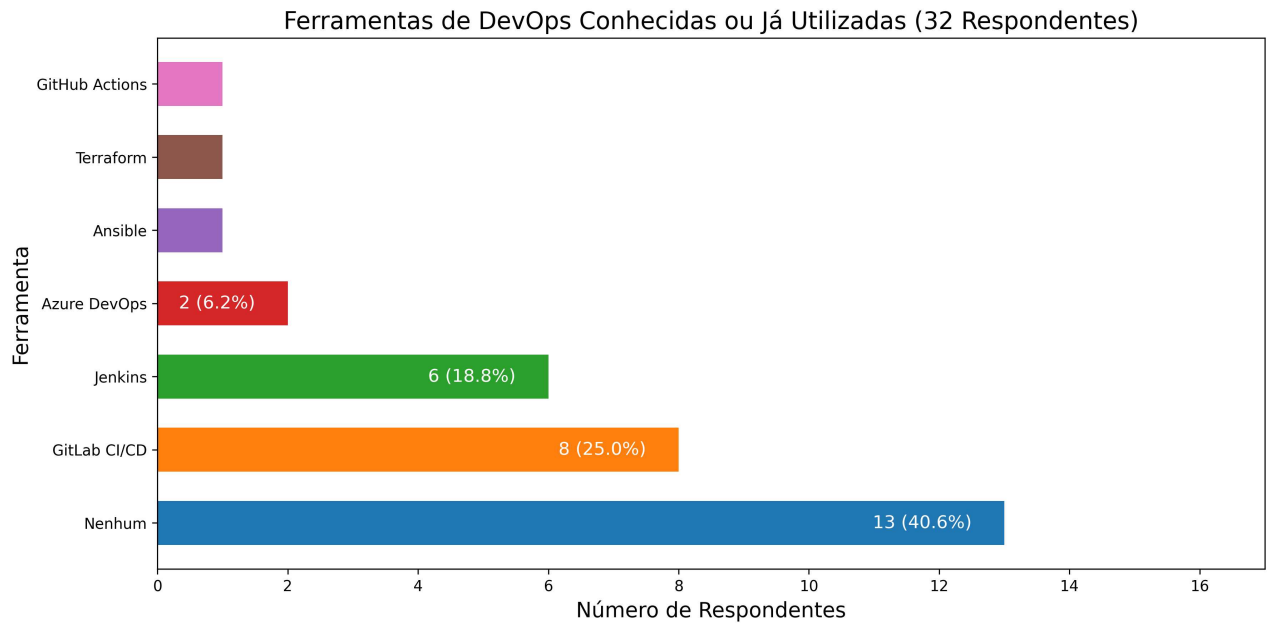
complexos e a inovação. Essa mudança não só melhora a velocidade do desenvolvimento, mas também ajuda a reduzir erros e inconsistências entre as etapas do ciclo de vida do software, promovendo uma entrega mais estável e confiável.

A **segurança nos processos**, embora mencionada de forma menos frequente, mostra que as práticas DevOps também são vistas como uma forma de mitigar riscos. A implementação de pipelines automatizados pode, por exemplo, incluir etapas de verificação de segurança, aumentando a confiabilidade do código entregue. Isso reforça que, mesmo em um ambiente de entregas rápidas e contínuas, a segurança não é negligenciada, mas sim integrada como parte do processo.

Outro benefício importante observado foi o impacto na **velocidade e eficiência**, características que se relacionam diretamente com a adoção das práticas de automação. Ao automatizar etapas críticas, como testes e deploys, as equipes conseguem entregar resultados com maior rapidez, atendendo melhor às necessidades dos clientes e stakeholders. Esse benefício está atrelado à ideia de que o DevOps permite uma resposta ágil e eficaz às demandas do mercado, criando um ciclo de desenvolvimento mais dinâmico.

Além disso, a **colaboração e comunicação no time** foi destacada como essencial

Figura 15 – Questionário sobre quais ferramentas de DevOps os participantes conhecem ou já utilizaram



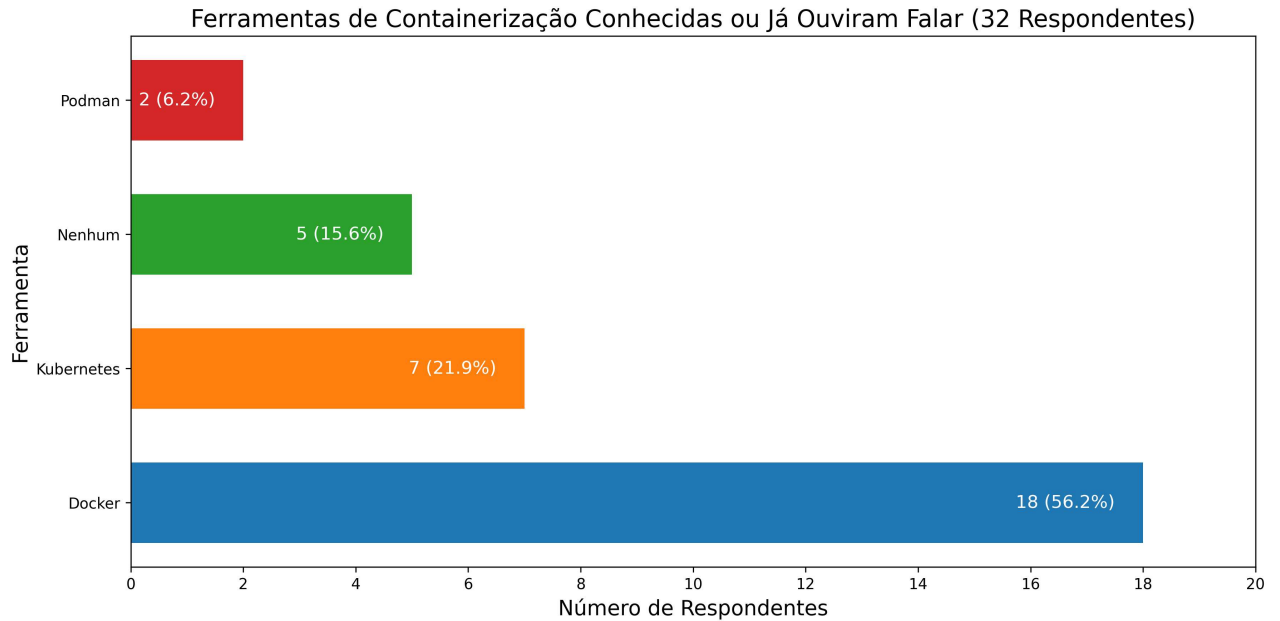
Fonte: Elaborado pelo autor (2025).

para a implementação bem-sucedida do DevOps. A integração das equipes de desenvolvimento e operações, facilitada pela automação e pela transparência dos processos, resulta em uma colaboração mais fluida. Isso fortalece o trabalho em equipe e promove um ambiente mais colaborativo, onde todos os membros estão alinhados com os objetivos comuns e podem atuar de forma mais coordenada.

Por fim, os benefícios apontados na **qualidade do código** e na **escalabilidade e confiabilidade** refletem ganhos de longo prazo. A adoção de práticas DevOps contribui para a criação de sistemas mais robustos, permitindo uma escalabilidade mais eficiente à medida que a base de usuários cresce. Além disso, a entrega contínua e a automação de testes ajudam a garantir que o código seja testado constantemente, o que melhora a qualidade do produto final.

No geral, os resultados sugerem que, embora o DevOps seja amplamente reconhecido por sua capacidade de otimizar processos e acelerar entregas, ele também desempenha um papel crucial na melhoria da segurança, colaboração e qualidade do desenvolvimento de software. Essa análise mostra que, ao adotar DevOps, as organizações não apenas melhoram a eficiência operacional, mas também criam um ambiente mais seguro e colaborativo, com impacto positivo no produto final.

Figura 16 – Questionário sobre quais ferramentas de containerização os participantes conhecem ou já ouviram falar



Fonte: Elaborado pelo autor (2025).

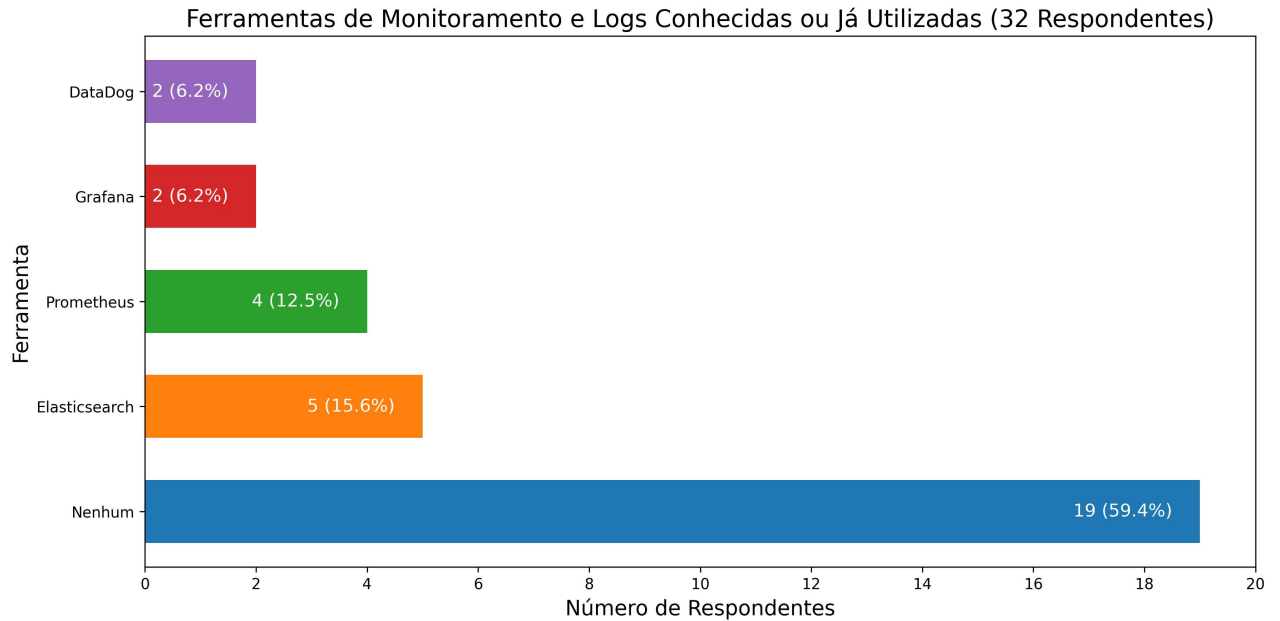
6.6 Maiores lacunas de conhecimento relacionadas a DevOps

Na Figura 19, apresentam-se as principais áreas que necessitam de melhoria para facilitar a adoção das práticas DevOps, proporcionando uma visão clara das áreas que exigem ajustes para otimizar o processo de implementação.

Em relação às áreas que precisam de melhorias para facilitar a adoção de práticas DevOps, o gráfico mostra que a **área de Desenvolvimento** foi a mais destacada pelos participantes, recebendo 13 votos, o que representa **40,6%** do total. Em segundo lugar, a **área de Qualidade & Desenvolvimento** foi mencionada por 7 participantes (**21,9%**). Além disso, as áreas de **Cultura e Colaboração** e **Treinamento e Capacitação** também receberam destaque, com 6 votos cada, correspondendo a **18,8%**. Esses dados apontam para a necessidade de avanços não apenas no aspecto técnico, mas também na promoção de uma cultura colaborativa e no investimento em capacitação, fatores essenciais para criar um ambiente que favoreça a adoção eficaz das práticas DevOps.

A análise qualitativa dos fatores que impactam a frequência de deploys revela desafios e prioridades cruciais para as equipes de desenvolvimento no processo de entrega de software. Entre os principais fatores identificados, destaca-se o "Tamanho do Projeto" como a maior barreira para a frequência de deploys. Projetos maiores e mais complexos demandam mais tempo e

Figura 17 – Questionário sobre quais ferramentas de monitoramento e logs os participantes conhecem ou já ouviram falar



Fonte: Elaborado pelo autor (2025).

recursos, tornando difícil manter um ritmo acelerado de entrega sem comprometer a qualidade ou estabilidade do sistema. Isso indica a necessidade de estratégias eficazes de modularização e escalabilidade para mitigar esses impactos.

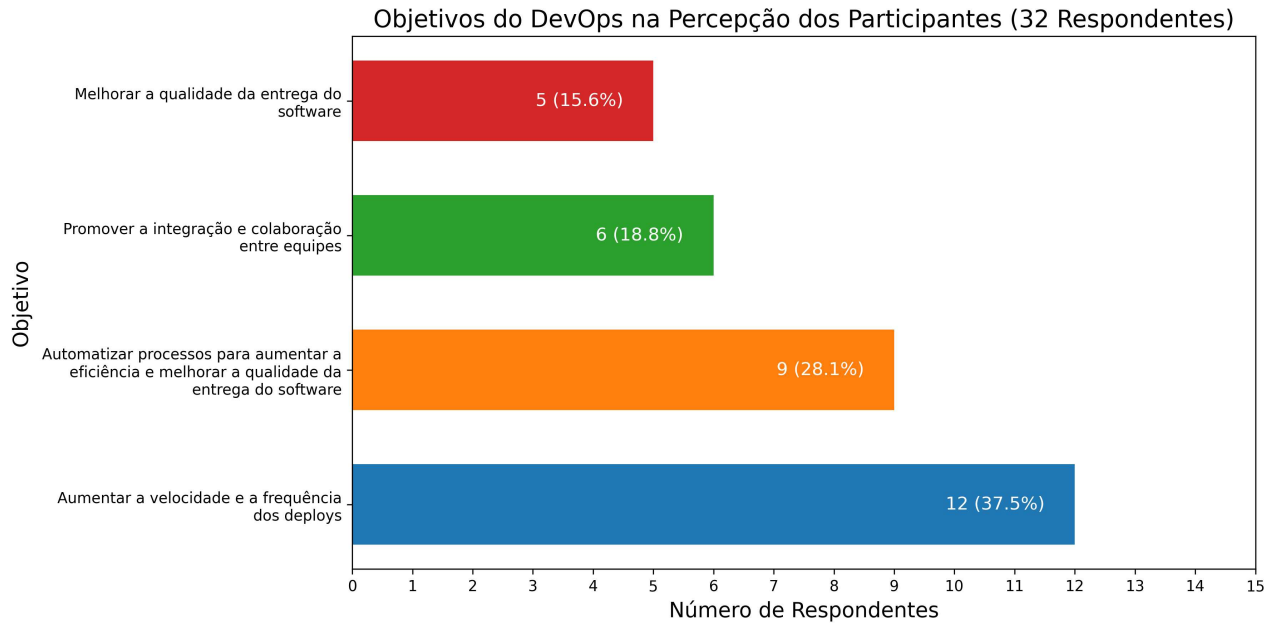
A "Automatização e o Uso de Ferramentas de CI/CD" surgem como elementos fundamentais para a aceleração do ciclo de entrega. Equipes que investem na automação de tarefas como build, testes e deploy relatam maior eficiência e segurança nos processos. A implementação de pipelines bem estruturados reduz gargalos operacionais e permite um fluxo contínuo de atualizações, minimizando a intervenção manual e mitigando erros humanos.

Outro aspecto relevante é a "Implementação de Novas Funcionalidades". A necessidade constante de aprimorar produtos e atender às demandas do mercado impulsiona a frequência de releases. No entanto, isso também exige um planejamento cuidadoso para garantir que as entregas rápidas não resultem em débito técnico ou regressão na estabilidade do sistema.

A "Cultura Organizacional" é um fator decisivo para o sucesso na adoção de DevOps. Empresas que promovem a colaboração entre equipes de desenvolvimento e operações conseguem maior alinhamento e eficiência nas entregas. Equipes que adotam uma mentalidade ágil, com feedback contínuo e melhoria incremental, são mais propensas a obter sucesso na implementação de processos DevOps.

Embora "Segurança" e "Testes" tenham sido mencionados com menor frequência,

Figura 18 – Questionário percepção dos participantes respeito dos principais objetivos do DevOps



Fonte: Elaborado pelo autor (2025).

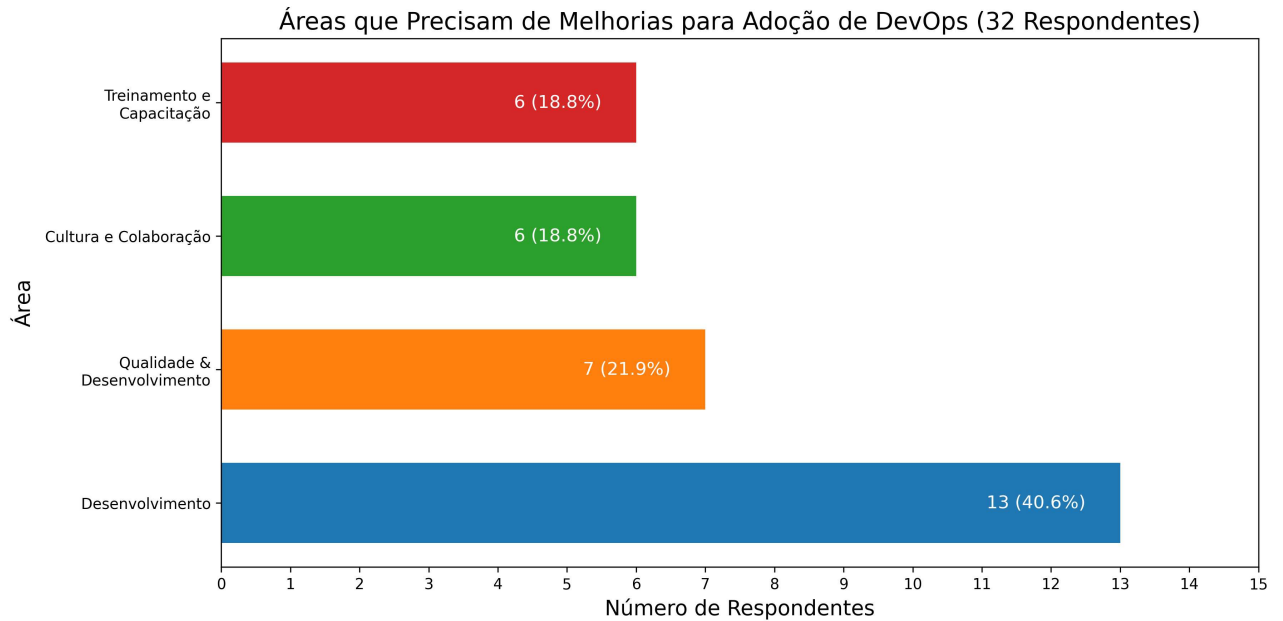
esses elementos continuam sendo essenciais para a qualidade e confiabilidade das entregas. O uso de testes automatizados e a implementação de boas práticas de segurança devem ser prioridades, mesmo em um ambiente de deploys frequentes.

Em resumo, a frequência de deploys é impactada por uma combinação de fatores técnicos e culturais. A automação, a gestão eficiente de projetos e uma cultura organizacional alinhada são fundamentais para equilibrar velocidade e qualidade na entrega de software. Equipes que conseguem integrar esses fatores de forma estratégica tendem a ter mais sucesso na adoção de DevOps.

6.7 Dificuldades enfrentadas por desenvolvedores com pouca experiência na adoção de DevOps

A análise dos desafios enfrentados pelos participantes na implementação das práticas DevOps revela aspectos críticos que impactam diretamente o sucesso da adoção dessa abordagem. O maior desafio identificado foi a **falta de conhecimento e treinamento**. Essa dificuldade é um reflexo da escassez de capacitação específica sobre DevOps dentro das organizações, o que é um obstáculo comum em ambientes onde as práticas tradicionais de desenvolvimento e operações predominam. A falta de treinamento formal pode resultar em uma adoção fragmentada e ineficaz,

Figura 19 – Questionário sobre as principais áreas que necessitam de melhoria para facilitar a adoção das práticas DevOps



Fonte: Elaborado pelo autor (2025).

dificultando a utilização plena das ferramentas e metodologias DevOps. No entanto, alguns participantes superaram essa barreira por meio de estratégias alternativas, como treinamentos informais, grupos de estudo e reuniões de tira-dúvidas. Isso sugere que, embora o treinamento formal seja desejável, as equipes estão encontrando maneiras de mitigar a falta de capacitação, o que pode ser um indicativo de resiliência organizacional e desejo de adaptação, essenciais para a transição para o DevOps.

Outro desafio importante foi a **divisão entre desenvolvimento e operações**, que continua a ser um obstáculo significativo para muitas organizações. A análise das respostas sugere que essa divisão não é apenas técnica, mas também cultural. A separação das equipes cria barreiras para a colaboração, comunicação e integração, essenciais para a implementação eficaz das práticas DevOps. O DevOps exige uma mentalidade de trabalho colaborativo, em que as equipes de desenvolvimento e operações trabalham juntas desde as fases iniciais do ciclo de vida do software. No entanto, as respostas indicam que muitas vezes as equipes ainda operam de maneira isolada, com objetivos e processos distintos. Essa falta de integração é frequentemente apontada como uma das maiores dificuldades para uma adoção bem-sucedida de DevOps.

Superar essa divisão, portanto, não envolve apenas mudanças nas ferramentas e processos, mas também uma transformação cultural dentro da organização.

Esses desafios – a falta de conhecimento e a divisão entre desenvolvimento e opera-

ções – foram destacados como os maiores obstáculos na adoção das práticas DevOps. A análise qualitativa sugere que, para que a implementação do DevOps seja bem-sucedida, é fundamental abordar essas questões de forma integrada.

A capacitação contínua e a transformação cultural são essenciais para que as equipes possam trabalhar de maneira colaborativa e utilizar as práticas e ferramentas DevOps de forma eficiente. Isso reforça a ideia de que o sucesso do DevOps depende não apenas de aspectos técnicos, mas também de um alinhamento cultural e organizacional, que deve ser priorizado pelas empresas que buscam adotar essa abordagem.

De maneira geral, os resultados obtidos com a aplicação do questionário evidenciam que, apesar de uma crescente conscientização sobre DevOps no ambiente acadêmico, a experiência prática com suas ferramentas e processos ainda é limitada entre os participantes. A maioria já ouviu falar do conceito, mas poucos possuem vivência real na aplicação de práticas como CI/CD, automação e containerização. Além disso, a frequência de deploys varia bastante entre as equipes, sendo influenciada por fatores como tamanho do projeto, automação e cultura organizacional. A análise também revelou que ferramentas como GitLab CI/CD e Docker são as mais conhecidas, embora muitas ainda sejam pouco exploradas. Esses achados reforçam a importância de iniciativas educacionais e práticas que incentivem a adoção de DevOps no contexto acadêmico, preparando melhor os alunos para desafios do mercado.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusão

Os resultados indicam que, embora o DevOps seja amplamente reconhecido como um facilitador da eficiência e qualidade nos processos de desenvolvimento de software, a implementação bem-sucedida ainda enfrenta desafios significativos. Entre as principais barreiras, destacam-se a falta de treinamento adequado, a resistência cultural dentro das equipes e a necessidade de adaptação das ferramentas e práticas ao contexto específico de cada organização. Além disso, a colaboração entre as áreas de desenvolvimento e operações é essencial para o sucesso da transformação digital. As sugestões levantadas pelos participantes reforçam a importância de investir em capacitação contínua, no apoio das lideranças e no fornecimento de recursos adequados às equipes, para que a integração do DevOps se torne uma prática eficaz e sustentável ao longo do tempo.

7.2 Considerações Finais

Este estudo revelou que, embora o DevOps seja amplamente reconhecido como uma abordagem eficaz para otimizar o desenvolvimento de software, sua adoção em projetos acadêmicos ainda enfrenta desafios importantes. Os dados analisados mostraram que a falta de conhecimento aprofundado, a resistência inicial à mudança e a dificuldade de adaptação das ferramentas ao ambiente universitário são barreiras frequentes para os estudantes. No entanto, a comparação com práticas consolidadas no mercado indica que, mesmo diante dessas dificuldades, os estudantes percebem benefícios claros na adoção do DevOps, como maior automação, melhor colaboração entre os membros da equipe e uma entrega mais ágil de funcionalidades. Com base nessas conclusões, este trabalho sugere algumas recomendações para tornar a adoção do DevOps mais acessível no meio acadêmico. Entre elas, destaca-se a importância de treinamentos práticos, a introdução de ferramentas DevOps desde as fases iniciais dos projetos e um maior apoio institucional para viabilizar ambientes de desenvolvimento mais alinhados às práticas do mercado. A implementação dessas sugestões pode ajudar os estudantes a desenvolver habilidades essenciais para o mundo profissional, promovendo uma cultura DevOps mais forte e preparando-os melhor para os desafios da área.

7.3 Trabalhos Futuros

Esta pesquisa trouxe uma visão inicial sobre a adoção do DevOps em projetos acadêmicos, destacando desafios, benefícios e impactos dessa abordagem. No entanto, há diversas oportunidades para aprofundamento e expansão desse estudo.

Uma possibilidade interessante para investigações futuras seria a realização de estudos de caso longitudinais, acompanhando a implementação do DevOps ao longo do tempo em diferentes projetos acadêmicos. Esse acompanhamento permitiria entender como a maturidade DevOps evolui nesses ambientes, identificar fatores críticos para o sucesso e mapear os desafios enfrentados em cada fase da adoção. Diante das dificuldades identificadas, algumas áreas de melhoria foram destacadas para tornar a adoção do DevOps mais eficiente e alinhada às necessidades dos projetos acadêmicos. Um dos pontos mais relevantes é a **capacitação e treinamento**, reforçando a importância de oferecer cursos, workshops e materiais educativos para que estudantes e professores possam desenvolver as habilidades essenciais para aplicar DevOps de forma eficaz. Além disso, pesquisas futuras poderiam explorar **ferramentas, metodologias e frameworks específicos** que facilitem a implementação do DevOps no contexto acadêmico. A criação de diretrizes estruturadas ou modelos de adoção adaptados à realidade universitária poderia auxiliar equipes de estudantes e docentes a incorporarem essas práticas de maneira mais eficiente em seus projetos. Outro aspecto fundamental a ser analisado é o **papel da cultura organizacional e da gestão de projetos acadêmicos** na adoção do DevOps. A forma como as universidades organizam seus fluxos de trabalho, gerenciam mudanças tecnológicas e incentivam práticas colaborativas pode impactar diretamente o sucesso da implementação dessa abordagem. Com essas perspectivas, espera-se que futuras pesquisas contribuam para uma adoção mais estruturada e eficiente do DevOps em projetos acadêmicos. Isso não apenas beneficiaria a formação dos alunos, preparando-os melhor para o mercado de trabalho, mas também elevaria a qualidade e a sustentabilidade das soluções desenvolvidas no ambiente universitário.

REFERÊNCIAS

- AHMAD, S.; KHAN, A.; ABBAS, H.; ALJOHANI, N. Challenges and barriers in software development: A systematic literature review. **IEEE Access**, IEEE, v. 9, p. 29698–29712, 2021.
- ALLSPAWE, J.; HAMMOND, P. 10+ deploys per day: Dev and ops cooperation at flickr. In: **Conferência Velocity da O’Reilly**. [S. l.: s. n.], 2009.
- DEBOIS, P. **Agile Infrastructure and Operations: How Infra-gile Are You?** S.l.: s.n., 2008. 202–207 p. Citado na página 15.
- FREITAS, P. H. d. S. Práticas de devops na melhoria da qualidade no desenvolvimento de software. Universidade Tecnológica Federal do Paraná, 2018.
- Grafana Labs. Grafana documentation. 2023. Disponível em: <https://grafana.com/docs/>. Acesso em: 24 jul. 2025.
- HUMBLE, C.; FARLEY, D. Implementação contínua: Melhores práticas e desafios. **Revista de Gerenciamento de Projetos de Software**, v. 8, n. 1, p. 30–42, 2015.
- JONES, M.; SMITH, L. Práticas e princípios de devops: Uma análise abrangente. **International Journal of Software Engineering**, Software Engineering Society, v. 22, n. 3, p. 134–150, 2019.
- KIM, G.; BEHR, K.; SPAFFORD, G. **The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win**. Portland: IT Revolution Press, 2013.
- KUMPULAINEN, V. Automation and improvement of software development with gitlab ci pipelines. 2023.
- LEVITA; NETO, J. A. M. Monitoramento contínuo de desempenho em ambientes de devops. **Journal of DevOps Engineering**, v. 3, n. 2, p. 55–68, 2018.
- NALLAMALA. **The do’s and don’ts of infrastructure code: A systematic gray literature review**. 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584921000720>. Acesso em: 24 jul. 2025.
- PAVAN, D. R.; SOUZA, E. F. d.; OLIVEIRA, R. A. P. Detectando e propondo melhorias em cenário devops de uma empresa de desenvolvimento de software. **COENS – Universidade Tecnológica Federal do Paraná**, Estr. p/ Boa Esperança, S/n - Zona Rural, Dois Vizinhos – PR – Brasil; Avenida Alberto Carazzai, 1640 – Centro – Cornélio Procopio, PR – Brasil, 2018.
- PETERS, K. S. Gitlab ci/cd: Efficient pipelines and devops practices. **Journal of Cloud Computing**, 2020. Disponível em: <https://link.springer.com/article/10.1186/s13677-020-00206-6>. Acesso em: 24 jul. 2025.
- PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. [S. l.]: McGraw-Hill, 2011.
- SHAH, A. Continuous integration and continuous delivery: A comprehensive review. **Software Engineering Review**, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0076831918300268>. Acesso em: 24 jul. 2025.

SHARMA, X.; COYNE, Y. Implementação de entrega contínua em ambientes de desenvolvimento Ágil. **Journal of Agile Engineering**, v. 7, n. 2, p. 112–125, 2015.

SILVA, N.; OLIVEIRA, M.; SANTOS, P. Título do artigo. **Revista Eletrônica de Sistemas de Informação**, Volume, n. Número, p. Página Inicial–Página Final, 2022. Disponível em: <https://sol.sbc.org.br/index.php/eres/article/view/13738>. Acesso em: 24 jul. 2025.

SOUSA, L. F. R. **DevOps: estudo de caso**. Tese (Doutorado), 2019. Disponível em: https://comum.rcaap.pt/bitstream/10400.26/31932/1/Leandro_sousa.pdf. Acesso em: 24 jul. 2025.

TAN, C. **Docker in Practice: Containerization and Orchestration**. [S. l.]: O'Reilly Media, 2021. Disponível em: <https://www.oreilly.com/library/view/docker-in/9781492059301/>. Acesso em: 24 jul. 2025.

VALENTE. **VIRTUALIZAÇÃO DE SERVIDORES UTILIZANDO PROX-MOX APLICADO AO AMBIENTE DEVOPS**. 2023. Disponível em: <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/8029>. Acesso em: 24 jul. 2025.