

Práticas de MLOps em softwares reais

Maurício Moraes Preto Carvalho¹, Sofia Larissa da Costa Paiva¹

¹Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74001-970 – Goiânia – Goiás – Brazil

moraesmauricio@discente.ufg.br, sofialarissa@ufg.br

Abstract. *In a scenario in which Machine Learning (ML) models are becoming more robust, there is a need for techniques that increase competitiveness among companies offering ML models as a service. Among these emerging techniques, Machine Learning Operations (MLOps), derived from Development and Operations (DevOps) practices, has stood out for automating the ML model life cycle, bringing greater practicality and reducing process bottlenecks associated with manual implementation. This paper presents the results of an exploratory study that investigated how MLOps has been used in practice. A study of practitioners in the field was conducted to understand the main stages and challenges of MLOps. The results guided an analysis of an MLOps process within a company. The results highlight the advantageous aspects, especially in the deployment and monitoring phases of ML projects, bringing the efficiency that is inherent in automation.*

Resumo. *Em um cenário no qual os modelos de Machine Learning (ML) vêm se tornando mais robustos, surge a necessidade de técnicas que aumentem a competitividade entre empresas que oferecem modelos de ML como um serviço. Entre essas técnicas emergentes, o Machine Learning Operations (MLOps), derivado das práticas de Desenvolvimento e Operações (DevOps), tem se destacado por automatizar o ciclo de vida dos modelos de ML para trazer maior praticidade e reduzir os gargalos do processo quando são realizados manualmente. Este artigo apresenta um estudo exploratório que investigou como MLOps tem sido utilizada na prática. Um estudo com praticantes da área foi realizado para entender as principais etapas e desafios de MLOps. Os resultados guiaram uma análise de um processo de MLOps em uma empresa. Os resultados destacam aspectos vantajosos, especialmente nas fases de implantação e monitoramento de projetos de ML, trazendo a eficiência que é inerente à automação.*

1. Introdução

Os modelos de Aprendizado de Máquina (do inglês, *Machine Learning* - ML) evoluíram significativamente ao longo dos anos e estão sendo amplamente utilizados em sistemas de software. Nesse cenário, a automação do ciclo de vida de ML tem ocorrido por meio das *Machine Learning Operations* (MLOps), que são um conjunto de práticas originadas do DevOps (Desenvolvimento e Operações) e adaptadas às particularidades de modelos de ML, permitindo entrega contínua e integrada [Mäkinen et al. 2021].

Muitas organizações ainda não tratam os modelos de aprendizado de máquina como softwares que devem seguir padrões de engenharia. Isso ocorre porque os projetistas

de ML enfrentam inúmeros dilemas para definir essas normas sob a ótica da Engenharia de Software. Por ser uma área que cresceu rapidamente no mercado, os modelos de ML frequentemente sofrem com a falta de padronização [SILVA et al. 2023], o que resulta em cerca de 22% das empresas que obtêm êxito na implantação de seus modelos em produção, de acordo com a Algorithmia [KUMARA et al. 2022]. O MLOps, então, surge como uma opção que viabiliza a padronização de ML, evitando desperdício de tempo e de capital, os quais, por serem elevados, podem levar à paralisação da implementação desses projetos. Por isso, compreender como aplicar o MLOps é essencial para garantir sua utilização de forma adequada.

Este estudo explorou como tem ocorrido a aplicação do MLOps na prática para superar os problemas relacionados ao gerenciamento do ciclo de vida de modelos de ML, analisando a literatura e as tecnologias existentes. Assim, um estudo com praticantes foi realizado para entender os pontos críticos no processo de MLOps. Os resultados guiaram a condução de uma análise da prática de MLOps em uma organização, identificando como os pontos críticos podem ser tratados de forma adequada, garantindo que seus modelos de ML sejam confiáveis, escaláveis e alinhados às necessidades do negócio e de seus *stakeholders*.

2. Referencial Teórico

O ciclo de vida do desenvolvimento de software (SDLC), criado nos anos 1960, compreende etapas contínuas da vida do software, como planejamento, desenvolvimento, testes, implantação e manutenção, assegurando rastreabilidade e evolução constante. Essas fases se repetem a cada nova entrega (*release*), permitindo identificar e aperfeiçoar pontos do processo. Tradicionalmente, o SDLC se dividia entre Desenvolvimento e Operação, o que gerava *silos* entre as equipes. As práticas de DevOps surgiram para integrar essas duas áreas, promovendo uma cultura de agilidade, colaboração e eficiência, além de auxiliar as equipes a enfrentar, de forma unificada, desafios relacionados ao desempenho, segurança e escalabilidade. Grandes empresas, como Netflix e Spotify, adotam essa cultura para consolidar processos e aprimorar os resultados de seus produtos [VALENTE 2020].

As principais práticas de DevOps incluem [OLIVEIRA 2018]:

- *Pipeline de Implantação*: processo automatizado que organiza as etapas de entrega do software (*Continuous Integration* - CI), permitindo a automação de testes, a integração contínua e o provisionamento de infraestrutura.
- *Entrega Contínua* (*Continuous Delivery* - CD): disponibiliza o software de forma automatizada, confiável e previsível.
- *Testes Contínuos e Automatizados*: executa automaticamente os testes a cada nova entrega, incluindo funcionalidades antigas.
- *Infraestrutura como Código* (*IaC*): gerencia e automatiza a infraestrutura de TI por meio de código, tornando processos manuais seguros, escaláveis e repetíveis.

A adoção do DevOps não exige a implementação imediata de todas as práticas e ferramentas. O mais importante é compreender os processos internos da organização e identificar como essa abordagem pode resolver os possíveis obstáculos existentes no ciclo de vida do software. Assim, os líderes de projeto devem avaliar cuidadosamente o modelo de desenvolvimento utilizado e priorizar os aspectos mais críticos.

Assim como o DevOps, o MLOps surge para transformar o ciclo de vida dos modelos, promovendo uma cultura colaborativa entre a equipe multidisciplinar que atua em todas as fases desse ciclo. Seu objetivo é otimizar essas etapas por meio de soluções e metodologias específicas aplicadas a esses projetos [KUMARA et al. 2022]. O desenvolvimento de modelos de ML é iterativo e exploratório, com etapas repetidas para buscar o melhor desempenho, o que evidencia a importância da automação [AMORIM 2022], conforme apresentado na Figura 1.

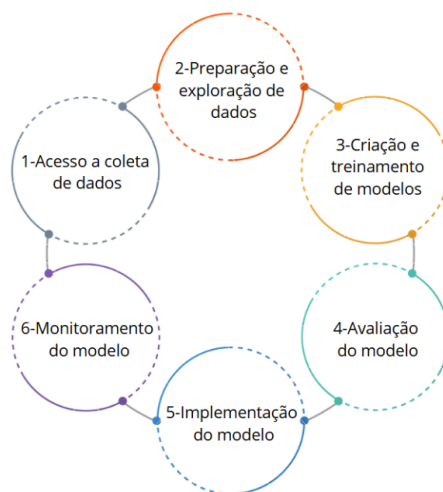


Figura 1. Ciclo de vida dos modelos de ML [ORACLE 2019]

Essas etapas sintetizam uma visão geral dos aspectos essenciais para a construção de ML, viabilizando a automação dos processos que ocorrem ao longo desse ciclo e de sua manutenção. A automação das etapas do ciclo de vida de modelos de ML é especialmente importante em empresas e startups, destacando-se em um cenário global de intensa competitividade, no qual a capacidade de entregar soluções de forma rápida e eficiente constitui um diferencial estratégico [Mäkinen et al. 2021].

Os modelos de ML enfrentam obstáculos operacionais significativos, principalmente em relação à manutenção e continuidade durante seu ciclo de vida em ambientes de produção. A falta de especialistas em MLOps aprofunda essas dificuldades, criando gargalos que comprometem a eficiência operacional. Nesse cenário, observa-se um desafio significativo em empresas cujo principal produto são os modelos de ML: a carência de iniciativas voltadas à automação do ciclo de vida desses modelos e da infraestrutura que os sustenta [AMORIM 2022].

2.1. Ferramentas para MLOps

Existem diversas ferramentas que podem ser utilizadas para o ciclo de vida de ML. A seguir, é apresentada uma comparação entre as principais ferramentas existentes no mercado, considerando os critérios propostos pela Oracle [ORACLE 2019] e recomendados pelo Google [GOOGLE 2025]: versionamento de dados, integração contínua, treinamento, automatização de pipelines, orquestrador, monitoramento e *open source*. Os resultados dessa comparação estão na Figura 3. Os campos vermelhos indicam que o critério não foi atendido, os campos amarelos indicam que o critério é atendido parcial-

mente e os campos verdes indicam que o critério é atendido totalmente pela ferramenta. As ferramentas comparadas foram:

1. DVC (Data Version Control)¹: versiona dados e modelos de ML, garantindo reprodutibilidade, rastreamento de alterações e integração com pipelines de CI/CD.
2. WML (Watson ML)²: gerencia todo o ciclo de vida de modelos de IA, oferecendo criação, implantação, monitoramento e verificação de imparcialidade e interpretabilidade, com opções *no-code* e para desenvolvedores.
3. CML (Continuous ML)³: automatiza pipelines de CI/CD para ML, realizando testes, treinamento, comparação de experimentos e monitoramento de dados, fornecendo feedback contínuo aos desenvolvedores.
4. Kubeflow⁴: apoia o ciclo de vida de IA, permitindo usar apenas os componentes necessários, como treinamento ou implantação de modelos, sem exigir a adoção completa da solução.
5. MLflow⁵: auxilia no gerenciamento do ciclo de vida de projetos de ML, tornando cada etapa mais controlável, auditável e reprodutível.
6. Amazon SageMaker⁶: cria, treina e implanta modelos de ML rapidamente, com interface gráfica e integração a IDEs, oferecendo recursos gratuitos para testes.
7. Comet⁷: gerencia o ciclo de vida de modelos de ML, permitindo monitoramento em tempo real, versionamento de dados e modelos, e integração com diversos frameworks, com versão gratuita limitada.
8. AML (Azure ML)⁸: treina, implanta e gerencia modelos de ML, com integração a frameworks populares. Não é gratuita mas possui opção de teste gratuito.
9. Pachyderm⁹: processa dados e cria pipelines acionados por alterações nos dados, oferecendo versionamento, reprodutibilidade, escalabilidade automática e integração com CI/CD.
10. Jenkins¹⁰: apoia CI/CD. Através dela é possível criar soluções com outras tecnologias, possibilitando testes, implementações e integrações por meio de automação com pipelines.
11. GitHubActions¹¹: cria fluxos de trabalho integrados a várias tecnologias, reagindo a eventos específicos, com plano gratuito limitado.

Portanto, o mercado oferece um ecossistema robusto de tecnologias para atender às demandas do ciclo de vida de modelos de ML, com opções pagas e gratuitas que permitem adequação a diferentes projetos. Cabe ao profissional de MLOps selecionar as ferramentas mais alinhadas aos objetivos de negócio. Além disso, muitos cientistas de dados também assumem a responsabilidade pela infraestrutura, o que reforça o interesse e a necessidade de práticas de MLOps. A adoção dessas práticas favorece a

¹<https://dvc.org/doc/start>

²<https://www.ibm.com/docs/en/software-hub/5.1.x?topic=services-watson-machine-learning>

³<https://cml.dev/doc/start>

⁴<https://www.kubeflow.org/docs/started/introduction/>

⁵<https://mlflow.org/docs/latest/>

⁶<https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html>

⁷<https://www.comet.com/site/>

⁸<https://azure.microsoft.com/en-us/products/machine-learning>

⁹<https://docs.pachyderm.com/products/mldm/latest/learn/key-features/>

¹⁰<https://www.jenkins.io/doc/>

¹¹<https://docs.github.com/pt/actions/get-started/understand-github-actions>

automação e a colaboração, permitindo que as equipes foquem na construção e otimização dos modelos, impulsionando maior maturidade no uso de ML no ambiente corporativo [Mäkinen et al. 2021].

Tecnologias	Versionamento de Dados	Integração Contínua	Treinamento	Automatização de Pipelines	Orquestrador	Monitoramento	Open Source
Data Version Control	✓	✓	✓
Watson Machine Learning	✓	✓	✓	✓	✓	✓	✗
Continuous Machine Learning	...	✓	...	✓	✓
KubeFlow	✓	✓	✓	...	✓
MLFlow	✓	✓	✓
Amazon Sage Maker AI	...	✓	✓	✓	✓	✓	✗
Comet	✓	✓	...	✗
Azure Machine Learning	...	✓	✓	✓	✓	✓	✗
Pachyderm	✓	✗
Jenkins	✗	✓	...	✓	✓	...	✓
GitHub Actions	✗	✓	...	✓	✓

Lenda ✓ Presente ... Parcial ✗ Ausente

Figura 2. Comparação entre ferramentas

3. Metodologia

O estudo teve caráter exploratório, na qual uma pesquisa de campo com praticantes identificou as principais etapas do processo de MLOps, as principais ferramentas e desafios envolvidos. Também, foi aplicado um questionário a profissionais que empregam as práticas de MLOps visando identificar as etapas essenciais, tecnologias utilizadas e perfil dos especialistas da área.

Os *insights* preliminares desse questionário foram utilizados como um guia para a análise do processo de MLOps praticado em uma empresa em um certo projeto. O projeto da empresa, que não será identificada, envolve uma solução de ML para automação e monitoramento de *call centers*. As questões a seguir, inspiradas em [Runeson e Höst 2009], foram utilizadas para mapear e definir o escopo junto à empresa: i) “Qual etapa de vida do modelo está sendo automatizada e por quê?”, ii) “Quais ferramentas, frameworks ou plataformas estão sendo usadas para automação do ciclo de vida do modelo?”, iii) “Quais os maiores desafios técnicos, organizacionais ou culturais encontrados na implementação de MLOps?” e iv) “Quais são os principais benefícios percebidos após a adoção de práticas de MLOps?”.

4. Resultados Preliminares

4.1. Questionário

O questionário ¹² foi aplicado a seis profissionais da área de MLOps, visando alinhar expectativas teóricas e práticas de mercado. Entre os respondentes, 2 pessoas tinham entre 26 e 30 anos, 4 pessoas possuíam mais de 5 anos de experiência, e os cargos incluíam

¹²https://docs.google.com/spreadsheets/d/1BL1MeRfrSE6Nr_gaLoEcPgfwFwk7VuUftbP198_sTQo/edit?usp=sharing

engenheiro de MLOps, arquiteto de solução e engenheiro de ML. As etapas mais valorizadas no ciclo de ML foram Preparação e exploração de dados e Acesso e coleta de dados, enquanto AWS e MLflow foram as ferramentas mais citadas, sendo o MLflow apontado com grande potencial mas com necessidade de melhorias em detecção de data drifts e documentação mais acessível. Os principais desafios mencionados foram garantir CI/CD para modelos e a escassez de profissionais qualificados. Os participantes indicaram que a adoção de MLOps depende da maturidade do projeto e da empresa, projetos exploratórios podem adotar práticas básicas, enquanto projetos de produção exigem práticas robustas para garantir colaboração eficiente e confiabilidade, com equipes geralmente contando com especialistas dedicados a MLOps.

4.1.1. Implantando MLOps

Considerando os resultados preliminares do questionário e da análise de adoção de MLOps na empresa pesquisada, a adoção do MLOps deve ser avaliada considerando a necessidade de entregar resultados mais rápidos e de valor aos usuários, reduzindo tensões organizacionais. Como a automação pode demandar tempo e recursos significativos, é essencial analisar sua real necessidade e, em cenários com restrições financeiras, de pessoal ou infraestrutura, priorizar a eliminação gradual de gargalos. Para garantir o sucesso da implementação, é fundamental promover colaboração entre equipes multidisciplinares, evitando silos de informação e retrabalho. A comunicação deve ser bem documentada e acessível, utilizando ferramentas que registrem e compartilhem informações de forma clara, assegurando alinhamento e transparência ao longo de todo o processo de MLOps.

As etapas a seguir apresentam um guia resumido de decisões a serem consideradas ao introduzir MLOps em projetos de ML: elaborado com base nas respostas à questão 10 do questionário:

- Etapa 1: Definir claramente o objetivo do projeto, identificando gargalos no ciclo de vida do modelo e o que pode ser automatizado para otimizar tempo e recursos. É essencial responder às perguntas “O que será resolvido?”, “Como?” e “Como será entregue?”. Um escopo bem definido é indispensável para avançar nas próximas etapas de implementação.
- Etapa 2: Criar pipelines para as rotinas do ciclo de vida do modelo, permitindo automatizar os principais gargalos e reduzir ou eliminar o trabalho manual nessas etapas.
- Etapa 3: Verificar se eles cumprem o esperado, garantindo que o gargalo identificado foi resolvido ou reduzido de forma significativa.
- Etapa 4: Monitorar o pipeline e o comportamento do modelo em operação, utilizando práticas de *CI/CD* para integração e *deploy*, com tecnologias específicas que serão detalhadas a seguir.
- Etapa 5: Verificar a sustentabilidade e o aprimoramento do sistema MLOps. É natural que surjam necessidades de manutenção e otimização. A pergunta-chave aqui é: “Necessita de manutenção?”.
 - Se a resposta for “Sim”: Indica que foram identificadas falhas, degradação de desempenho ou novas oportunidades de melhoria, como retreinamento de modelos, otimização de pipelines, atualização de bibliotecas ou expansão do escopo. Nesse caso, o processo retorna à Etapa 1, iniciando

- uma nova ”*sprint*” de desenvolvimento ou otimização, onde o novo problema ou oportunidade é reavaliado e incorporado ao escopo.
- Se a resposta for ”Não”: Indica que o sistema está operando conforme o esperado, os gargalos foram resolvidos e o desempenho é satisfatório. O processo avança para o ”Fim da *sprint*”, sinalizando que os objetivos foram alcançados e o sistema está estável, aguardando futuras avaliações ou novas necessidades.

A Figura 3 ilustra, em forma de fluxograma, os passos descritos anteriormente.

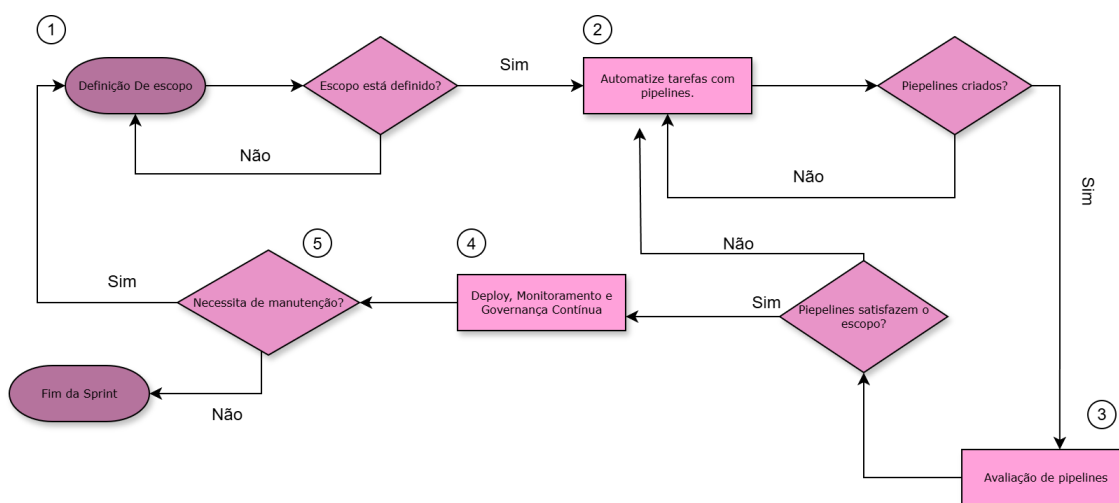


Figura 3. Fluxograma da implantação de MLOps

4.1.2. O profissional de MLOps

Essa seção mostra os resultados preliminares da questão 12 do questionário. O mercado de MLOps ainda é emergente e encontra resistência fora das grandes corporações, onde a demanda é crescente e exige profissionais com competências técnicas, habilidades interpessoais e, em alguns casos, certificações [KORADA 2023]. Para ingressar na área, é essencial ter base sólida em engenharia de software e práticas DevOps, incluindo proficiência em CI/CD, Python, monitoramento de sistemas, containerização, orquestração, infraestrutura como código e controle de versão, além de experiência com nuvem.

O profissional de MLOps atua como um integrador ao longo do ciclo de vida de modelos de ML, desde o treinamento até a mitigação de problemas como o drift de dados. Seu papel exige capacidade analítica para compreender problemas e escolher ferramentas adequadas, além de uma mentalidade colaborativa para conectar equipes de infraestrutura, engenharia de dados, ciência de dados e negócios, garantindo soluções contínuas, seguras e escaláveis.

4.2. Análise de projeto prático em empresa

O relato da empresa às questões de pesquisa trouxeram os resultados descritos a seguir. A maturidade em MLOps é evidente pela definição clara do escopo, alinhada aos passos críticos descritos na subseção 4.1.1, em um contexto de desafios significativos na entrega

de um produto dependente de modelos de ML. Para lidar com essa complexidade, foi essencial investir em colaboração e documentação centralizada. O uso do Notion integrou equipes multidisciplinares e evitou silos de informação, enquanto a mobilidade de profissionais entre times e a criação de um escritório virtual no Gather aumentaram a interação entre áreas, dinamizaram os rituais ágeis e aceleraram a validação de decisões técnicas.

Com a percepção da equipe alinhada, foram identificados dois gargalos principais: implementação e monitoramento, reconhecidos como os maiores entraves. O engenheiro de MLOps destacou a necessidade de entregas mais frequentes e ágeis, tornando a automação desses processos indispensável para a evolução do produto. Em vez de buscar um ambiente sem falhas (raro em sistemas reais), o foco passou a ser a construção de uma estrutura resiliente, capaz de detectar falhas rapidamente e restaurar serviços, garantindo estabilidade mesmo diante de quedas de desempenho ou *data drift*.

Com o escopo agora claramente definido, a equipe pôde avançar para o passo 2: a criação de pipelines para a automação das fases de implementação e monitoramento. Para garantir a rastreabilidade e a colaboração, o versionamento e o armazenamento das atualizações desses pipelines no GitHub foram padronizados, permitindo que toda a equipe pudesse validar e testar as modificações. Essa abordagem possibilitou a construção e rastreabilidade de pipelines capazes de gerar resultados significativos para o ciclo de vida da ML.

Durante essa fase, também foram discutidas e selecionadas as ferramentas ideais para essa automação. Assim, um pipeline de *CI/CD* foi criado com Jenkins, responsável pelo *deploy* do sistema. Após a conclusão das etapas do Jenkins, o GitHub Actions foi integrado para receber o controle do fluxo, sendo então responsável por configurar e instanciar a infraestrutura necessária na AWS utilizando Terraform. Convém destacar, ainda, que o Qase foi incorporado ao processo como ferramenta de gestão e execução de testes, atuando como um componente essencial para garantir a qualidade dos artefatos gerados em cada etapa do pipeline. Com sua integração ao GitHub Actions, os casos de teste eram executados automaticamente a cada nova versão do código, permitindo que a equipe de QA acompanhasse o desempenho dos testes em tempo real, identificasse rapidamente falhas e priorizasse correções. O Qase também serviu como repositório central para os planos e resultados de testes, facilitando a comunicação entre desenvolvedores e analistas de qualidade e garantindo que as modificações fossem entregues com sucesso.

Para gerenciar essa infraestrutura, o Terraform foi empregado para definir, versionar e provisionar os recursos da AWS. Isso englobou a configuração de serviços como AWS Elastic Container Service, utilizando o modo Fargate para a execução dos contêineres dos modelos sem a necessidade de gerenciar servidores subjacentes, e também instâncias EC2 para tarefas específicas que exigiam maior controle. Com o Terraform, a infraestrutura pôde ser replicada e atualizada de forma consistente e automatizada. Complementarmente, os pipelines (incluindo as partes orquestradas por Jenkins e GitHub Actions) foram integrados ao AWS CodePipeline para orquestrar o fluxo de entrega contínua e ao AWS Step Functions para coordenar tarefas complexas e fluxos de trabalho do ciclo de vida dos modelos, assegurando um *deploy* eficiente e um monitoramento contínuo da aplicação em produção.

Após a verificação rigorosa, foi confirmado que a automação estava realizando os

passos estipulados, concluindo com sucesso o passo 3. Atualmente, o passo 4 envolve o monitoramento contínuo do modelo e do comportamento dos pipelines de automação, garantindo que cumpram seus objetivos e não apresentem má formação de contêineres ou instabilidades nos ambientes criados pelas diferentes tecnologias e pipelines. Em outras palavras, a equipe assegura que o ambiente se mantenha estável e que os serviços programados sejam providos conforme o esperado. Até o momento, a automação implementada não apresentou nenhuma instabilidade, demonstrando sua robustez. Por fim, os resultados obtidos estão resumidos na Figura 4.

Questões de Pesquisa	Respostas
i) Qual etapa de vida do modelo está sendo automatizada e por quê?	Deploy e Monitoramento.
ii) Quais ferramentas, frameworks ou plataformas estão sendo usadas para automação do ciclo de vida do modelo?	Jenkins (fluxo de trabalho), Github Actions (fluxo de trabalho), Terraform (infraestrutura), Qase (teste integrado), AWS (infraestrutura), Notion (documentação), GitHub (versionamento), AWS CodePipeline e AWS Step Functions (orquestrar fluxo dentro do ambiente AWS).
iii) Quais os maiores desafios técnicos, organizacionais ou culturais encontrados na implementação MLOps?	Os maiores desafios na implementação de MLOps foram de natureza organizacional e cultural, focados principalmente em superar os silos de informação e promover a colaboração entre equipes. Para mitigar esses problemas, a empresa adotou estratégias como a alocação de profissionais multidisciplinares em diferentes áreas do projeto para facilitar o compartilhamento de conhecimento e a criação de um escritório virtual para dinamizar a interação e manter a coesão da equipe.
iv) Quais são os principais benefícios percebidos após a adoção de práticas de MLOps?	Houve uma diminuição drástica no ciclo de entrega, permitindo que tarefas que antes levavam longas sprints fossem concluídas em apenas um dia. Além disso, a automação demonstrou sua robustez e eficiência, garantindo que o ambiente se mantivesse estável. A melhoria na colaboração e na comunicação também foi um ponto crucial, pois a centralização da documentação e a abordagem colaborativa ajudaram a dissolver os silos de informação, garantindo o sucesso da solução.

Figura 4. Sumarização dos resultados

5. Conclusão

Os resultados preliminares deste estudo exploratório, que incluiu um questionário com profissionais, a definição do fluxograma para implantação de MLOps em projetos de ML e a análise prática em uma empresa, permitem compreender de forma mais ampla os desafios e benefícios envolvidos na adoção de práticas de MLOps em contextos reais.

O questionário forneceu informações valiosas que se refletiram diretamente na construção do fluxograma e, conseqüentemente, contribuíram para a compreensão e análise do caso prático. A aplicação do fluxograma permite identificar gargalos e validar melhorias especialmente nas etapas de implantação e monitoramento, consideradas as mais críticas. Além disso, o uso de ferramentas específicas reduz significativamente o tempo de entrega, possibilitando que processos sejam concluídos em apenas um dia.

O principal desafio identificado foi de natureza organizacional e cultural, decorrente da fragmentação entre equipes e da ausência de processos colaborativos consolidados. A centralização da documentação e a formação de times multidisciplinares mostraram-se estratégias eficazes para mitigar essas barreiras, fortalecendo a integração e consolidando o MLOps como prática estratégica dentro da empresa.

Essas constatações estão alinhados às conclusões de [Nogare e Silveira 2024], que também apontam ganhos expressivos de eficiência e automação na adoção de práticas de MLOps. Além disso, reforçam a importância de considerar fatores humanos e organizacionais como elementos centrais para o sucesso e a sustentabilidade dessas iniciativas em ambientes complexos.

Como trabalhos futuros, recomenda-se o aprofundamento em estudos de caso voltados a etapas específicas do ciclo de vida de modelos de aprendizado de máquina, a fim

de demonstrar, de forma prática e detalhada, os impactos da implantação de MLOps.

Referências

- [AMORIM 2022]AMORIM, B. F. Trabalho de Conclusão de Curso, *Uma investigação dos desafios no ciclo de vida do aprendizado de máquina e a importância do MLOps: um survey*. 2022. Trabalho de Conclusão de Curso (Especialização em Ciência de Dados) – Universidade Tecnológica Federal do Paraná, Campus Dois Vizinhos, Dois Vizinhos.
- [GOOGLE 2025]GOOGLE. *Engines Mlops List*. 2025. Acessado em: 31 jul. 2025. Disponível em: <<https://www.google.com/search?q=engines+mlops+listoq=engines+mlops+list>>.
- [KORADA 2023]KORADA, L. Aiops and mlops: Redefining software engineering lifecycles and professional skills for the modern era. *Journal of Engineering and Applied Sciences Technology*, p. 5, 2023. ISSN 2634-8853.
- [KUMARA et al. 2022]KUMARA, I. et al. Requirements and reference architecture for mlops: Insights from industry. *TechRxiv*, 2022. Preprint publicado em TechRxiv. Disponível em: <<https://www.techrxiv.org/users/687559/articles/680271-requirements-and-reference-architecture-for-mlops-insights-from-industry>>.
- [Mäkinen et al. 2021]MÄKINEN, S. et al. Who needs mlops: What data scientists seek to accomplish and how can mlops help? In: *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN) of 43rd International Conference on Software Engineering (ICSE)*. [S.l.: s.n.], 2021. p. 20–27.
- [Nogare e Silveira 2024]NOGARE, D.; SILVEIRA, I. F. Experimentation, deployment and monitoring machine learning models: Approaches for applying mlops. *arXiv preprint arXiv:2408.11112*, São Paulo, ago. 2024. Publicado em 22 de agosto de 2024. Disponível em: <<https://arxiv.org/abs/2408.11112>>.
- [OLIVEIRA 2018]OLIVEIRA, W. C. d. S. *Aplicando análise qualitativa para caracterização dos princípios DevOps em uma empresa do setor público*. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação), São Luís, 2018. Disponível em: <<http://hdl.handle.net/123456789/3470>>.
- [ORACLE 2019]ORACLE. *Data science lifecycle: um guia para a ciência de dados*. 2019. Disponível em: <<https://www.oracle.com/a/ocom/docs/data-science-lifecycle-ebook.pdf>>.
- [Runeson e Höst 2009]RUNESON, P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, Springer, v. 14, n. 1, p. 131–164, 2009. Disponível em: <<https://doi.org/10.1007/s10664-008-9102-8>>.
- [SILVA et al. 2023]SILVA, L. C. et al. Qualidade de software para engenheiros de ia: Um estudo inicial da realidade brasileira. *Universidade Estadual do Ceará*, Fortaleza, Ceará, 2023. Disponível em: <<https://sol.sbc.org.br/index.php/ise/article/download/26118/25941/>>.
- [VALENTE 2020]VALENTE, M. T. *Engenharia de Software Moderna*. 1. ed. [s.n.], 2020. ISBN 978-6500019506. Disponível em: <<https://engsoftmoderna.info/>>.