



João Pedro Pinho Duarte

DevOps and (Software) Operational Excellence Metrics

Coimbra, abril de 2025



João Pedro Pinho Duarte

DevOps and (Software) Operational Excellence Metrics

Relatório de estágio submetido ao Instituto Superior de Contabilidade e Administração de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação de Gestão, realizado sob a orientação do Prof. Doutor António Rui Trigo Ribeiro, a coorientação do Prof. Doutor Fernando Paulo dos Santos Rodrigues Belfo e a supervisão do Engenheiro Pedro Rafael Pimenta Saraiva.

Coimbra, abril de 2025

TERMO DE RESPONSABILIDADE

Declaro ser o autor deste relatório de estágio, que constitui um trabalho original e inédito, que nunca foi submetido a outra Instituição de ensino superior para obtenção de um grau académico ou outra habilitação. Atesto ainda que todas as citações estão devidamente identificadas e que tenho consciência de que o plágio constitui uma grave falta de ética, que poderá resultar na anulação do presente relatório de estágio.

PENSAMENTO

“Currently, DevOps is more like a philosophical movement, not yet a precise collection of practices, descriptive or prescriptive.” – Gene Kim

Esta declaração sintetiza com exatidão o carácter dinâmico do paradigma *DevOps* no atual contexto tecnológico.

AGRADECIMENTOS

A realização deste projeto, no âmbito do mestrado em Sistemas de Informação de Gestão, representou uma longa jornada repleta de desafios, conquistas e aprendizagens, que só foi possível concretizar graças ao apoio fundamental de várias pessoas.

Em primeiro lugar, gostaria de expressar a minha profunda gratidão aos meus pais, pelo incansável esforço em proporcionar-me a oportunidade de estudar e alcançar os meus objetivos. O meu sincero obrigado pela paciência demonstrada ao longo de todos estes anos e, sobretudo, pelo constante incentivo que me deram.

Um especial agradecimento ao meu orientador, Professor Doutor António Rui Trigo Ribeiro, e ao meu coorientador, Professor Doutor Fernando Paulo dos Santos Rodrigues Belfo, pela motivação, disponibilidade e valioso apoio prestado durante todo este processo.

Por último, mas não menos importante, gostaria de agradecer a toda a equipa da ANOVA, em particular ao meu supervisor, Engenheiro Pedro Rafael Pimenta Saraiva, pelo acompanhamento em todo o percurso, pela partilha do seu conhecimento de forma profissional e competente, e pela permanente disponibilidade demonstrada.

RESUMO

Nos últimos anos, o desenvolvimento e a entrega contínua de *software* têm vindo a ser profundamente impactados pelas práticas *DevOps*, que promovem a integração entre equipas de desenvolvimento e operações, com o apoio de ferramentas de automatização e monitorização. A necessidade crescente de entregar soluções mais rápidas, fiáveis e eficientes tem impulsionado a adoção desta abordagem por empresas de diversas áreas.

A ANOVA enfrentava o desafio de alcançar uma visão clara e objetiva dos seus processos de desenvolvimento de software, num contexto em que se procurava adotar práticas alinhadas com a cultura *DevOps*. A ausência de uma estrutura centralizada de monitorização comprometia a visibilidade sobre o desempenho das equipas, dificultando a identificação de problemas e a tomada de decisões informadas, tanto a nível técnico como de gestão.

O principal objetivo deste projeto de estágio consistiu no desenvolvimento de um *dashboard* automatizado em *Power BI*, capaz de apresentar, de forma clara e interativa, métricas relevantes do ciclo *DevOps* da organização. Para tal, recorreu-se à metodologia *Design Science Research*, permitindo criar um artefacto funcional que responde a um problema organizacional concreto. O projeto envolveu a recolha de dados, criação de pipelines de ETL, modelação multidimensional e definição de indicadores com DAX.

A implementação deste *dashboard* resultou numa melhoria significativa na capacidade de monitorização dos processos internos da ANOVA, potenciando a melhoria contínua, a identificação atempada de gargalos e uma cultura orientada a dados. Para além disso, esta ferramenta revelou-se fundamental na aproximação entre equipas técnicas e de gestão, permitindo uma leitura mais intuitiva e colaborativa do desempenho da organização.

Palavras-chave: Desenvolvimento de Software; DevOps, Métricas de DevOps, DevOps Dashboard, Power BI.

ABSTRACT

In recent years, the development and continuous delivery of software has been profoundly impacted by DevOps practices, which promote integration between development and operations teams, with the support of automation and monitoring tools. The growing need to deliver faster, more reliable and more efficient solutions has driven the adoption of this approach by companies in various fields.

ANOVA was faced with the challenge of achieving a clear and objective vision of its software development processes, in a context where it was seeking to adopt practices in line with the DevOps culture. The lack of a centralised monitoring structure compromised visibility of team performance, making it difficult to identify problems and make informed decisions, both at a technical and management level.

The main objective of this internship project was to develop an automated dashboard in Power BI, capable of presenting relevant metrics from the organisation's DevOps cycle in a clear and interactive way. To this end, the Design Science Research methodology was used to create a functional artefact that responds to a specific organisational problem. The project involved collecting data, creating ETL pipelines, multidimensional modelling and defining indicators with DAX.

The implementation of this dashboard resulted in a significant improvement in the monitoring capacity of ANOVA's internal processes, boosting continuous improvement, the timely identification of bottlenecks and a data-orientated culture. In addition, this tool has proved fundamental in bringing technical and management teams closer together, allowing for a more intuitive and collaborative reading of the organisation's performance.

Keywords: Software Development; DevOps, DevOps Metrics, DevOps Dashboard, Power BI.

ÍNDICE GERAL

INTRODUÇÃO.....	1
1 ESTADO DE ARTE.....	3
1.1 Origens do DevOps	3
1.1.1 Influências	3
1.1.2 Problemas dos Modelos Tradicionais.....	4
1.1.3 DevOps como Resposta.....	4
1.1.4 Definição Moderna de DevOps	5
1.2 Pilares do DevOps	5
1.3 Práticas Essenciais do DevOps.....	8
1.4 Ciclo do DevOps	11
1.5 Métricas de DevOps	12
1.5.1 Métricas DORA.....	14
1.5.2 Métricas de DevOps selecionadas	20
1.6 Utilização das métricas.....	26
2 METODOLOGIA.....	28
3 ENTIDADE ACOLHEDORA	32
3.1 Intelligent Sensing Anywhere, S.A.	32
3.2 ANOVA.....	33
4 PROJETO	35
4.1 Identificação do Problema.....	35
4.2 Definição dos Objetivos	36
4.3 Desenvolvimento do Artefacto.....	38

DevOps and (Software) Operational Excellence Metrics

4.3.1	Conceção	38
4.3.2	Implementação.....	46
4.4	Demonstração	61
4.5	Avaliação	66
4.5.1	Dashboards finais	69
4.5.2	Definição dos elementos visuais	71
4.6	Comunicação	73
CONCLUSÃO.....		75
REFERÊNCIAS		77
APÊNDICES		82
APÊNDICE 1. 24 PRÁTICAS DEVOPS		83

ÍNDICE DE TABELAS E FIGURAS

Tabela 1.1. Design vs. Entrega	15
Tabela 1.2. Software Delivery Performance	17
Tabela 2.1. Design science research principles	28
Tabela 4.1. Tabelas importadas do modelo semântico “DevopsMetrics”	47
Tabela 4.2. ID e designação dos dashboards	68
Tabela 4.3. ID, designação e identificação dos dashbaords a que pertencem os KPI	68
Tabela 4.4. Categorias de performance dos KPI	69
Figura 1.1. Pipeline de DevOps	12
Figura 2.1. Metodologia DSR	29
Figura 4.1. Desenho do modelo multidimensional no esquema em estrela	43
Figura 4.2. Artefacto a desenvolver em Power BI	44
Figura 4.3. Mockup para o dashboard “Frequência de Implementações (Deployment Frequency)”	45
Figura 4.4. Modelo Multidimensional	49
Figura 4.5. Exemplo de um extrato do código em DAX para contagem de implementações bem-sucedidas	53
Figura 4.6. Exemplo de fórmula em DAX para média de implementações por dia, utilizando a função COUNTA e divisão por período fixo (28 dias)	53
Figura 4.7. Medida DAX: Contagem de Bugs no Ambiente de Produção	54
Figura 4.8. Expressão DAX para contabilizar o número total de bugs reportados em ambientes críticos	55

Figura 4.9. Fórmula DAX para calcular o número de bugs específicos do ambiente de staging	57
Figura 4.10. Fórmula DAX para Taxa de Defeitos em Produção (%)	58
Figura 4.11. Filtros Temporais: análise com precisão cronológica.....	59
Figura 4.12. Filtros Organizacionais: segmentação por estrutura empresarial.....	59
Figura 4.13. Lista de áreas disponíveis	60
Figura 4.14. Botões de ação rápida do tipo marcador	60
Figura 4.15. Dashboard da Frequência de Implementações (Deployment Frequency).....	62
Figura 4.16. Dashboard da Taxa de Defeitos em Produção (Defect Escape Rate)	63
Figura 4.17. Tabela oculta presente no Dashboard da Frequência de Implementações (Deployment Frequency)	64
Figura 4.18. Tabela oculta presente no Dashboard da Taxa de Defeitos em Produção (Defect Escape Rate)	64
Figura 4.19. Dashboard do Tempo de Ciclo (Cycle Time)	65
Figura 4.20. Dashboard do Tempo de Espera (Lead Time)	65
Figura 4.21. Dashboard da Frequência de Implementações (Deployment Frequency), ajustada após avaliação.....	69
Figura 4.22. Dashboard da Taxa de Defeitos em Produção (Defect Escape Rate)	70
Figura 4.23. Dashboard do Tempo de Ciclo (Cycle Time)	70
Figura 4.24. Dashboard do Tempo de Espera (Lead Time)	71
Figura 4.25. Exemplo de um código para a condição das cores consoante o valor	72
Figura 4.26. Exemplo de um código para a condição das cores consoante o valor	72
Figura 4.27. Exemplo de um código para a condição das cores consoante o valor	73

Lista de abreviaturas, acrónimos e siglas

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
CAB	<i>Change Approval Board</i>
CALMS	<i>Culture, Automation, Lean, Measurement, Sharing</i>
CAMS	<i>Culture, Automation, Measurement, Sharing</i>
CD	<i>Continuous Delivery</i>
CI	<i>Continuous Integration</i>
DAX	<i>Data Analysis Expressions</i>
DevOps	<i>Development + Operations</i>
DORA	<i>DevOps Research and Assessment</i>
DSR	<i>Design Science Research</i>
DSRM	<i>Design Science Research Methodology</i>
ETL	<i>Extract, Transform, Load</i>
FDRT	<i>Failed Deployment Recovery Time</i>
HaaS	<i>Hardware as a Service</i>
IoT	<i>Internet of Things</i>
ISA	<i>Intelligent Sensing Anywhere, S.A.</i>
IT	<i>Information Technology</i>
KPI	<i>Key Performance Indicator</i>
MTTD	<i>Mean Time to Detection</i>
MTTR	<i>Mean Time to Recovery</i>
OAuth2	<i>Open Authorization 2.0</i>
OData	<i>Open Data Protocol</i>

DevOps and (Software) Operational Excellence Metrics

Ops	<i>Operations</i>
RLS	<i>Row-Level Security</i>
ROI	<i>Return on Investment</i>
RTU	<i>Remote Telemetry Unit</i>
SDLC	<i>Software Development Life Cycle</i>
SVN	<i>Subversion</i>
WIP	<i>Work In Progress</i>

INTRODUÇÃO

A aceleração digital e o aumento da complexidade nos processos de desenvolvimento de *software* conduziram à necessidade de novas abordagens que promovam maior colaboração, agilidade e automação. Neste contexto, o *DevOps* surge como um paradigma transformador, que procura unir equipas tradicionalmente isoladas — desenvolvimento e operações — com o intuito de garantir entregas mais rápidas, estáveis e alinhadas com os objetivos de negócio.

A adoção de práticas *DevOps* tem-se revelado essencial para organizações que desejam manter-se competitivas num mercado onde os ciclos de vida do software são cada vez mais curtos e exigentes. A integração de ferramentas de monitorização, automatização de *deploys*, testes contínuos e visualização de métricas de desempenho torna-se um fator crítico para a eficácia e eficiência das equipas.

A ANOVA, empresa de base tecnológica com forte enfoque em soluções de monitorização remota e IoT, enfrentava dificuldades em obter uma visão global e em tempo real dos seus processos de desenvolvimento. A inexistência de um sistema centralizado de métricas dificultava a avaliação da performance do *pipeline DevOps*, comprometendo a capacidade de resposta da organização a eventuais ineficiências.

Foi neste contexto que surgiu a oportunidade de desenvolver um projeto de estágio de mestrado integrado no curso de Sistemas de Informação de Gestão do Instituto Superior de Contabilidade e Administração de Coimbra, centrado na criação de um *dashboard* automatizado de métricas *DevOps*, com recurso à ferramenta *Power BI*. O objetivo foi conceber uma solução que permitisse visualizar, analisar e interpretar os dados extraídos das várias fases do ciclo de desenvolvimento, com vista à tomada de decisões mais fundamentadas e ao reforço da cultura de melhoria contínua.

A metodologia adotada para este projeto foi a *Design Science Research* (DSR), frequentemente utilizada para resolver problemas práticos através da construção e validação de artefactos tecnológicos. O desenvolvimento do *dashboard* incluiu etapas de levantamento de requisitos, modelação de dados, implementação de processos de Extração, Transformação

DevOps and (Software) Operational Excellence Metrics

e Carregamento (ETL, do inglês *Extract, Transform and Load*), cálculo de métricas com *Data Analysis Expressions* (DAX) e construção das visualizações interativas.

Este relatório encontra-se estruturado em seis capítulos: Introdução; Enquadramento Teórico; Metodologia; Entidade Acolhedora; Projeto (dividido em identificação do problema, desenvolvimento do artefacto, avaliação e comunicação); e Conclusão. O objetivo é apresentar uma visão abrangente e fundamentada sobre o processo de conceção, desenvolvimento e implementação da solução proposta, assim como refletir sobre o seu impacto e contributos para a organização e para a formação do estagiário.

1 ESTADO DE ARTE

O *DevOps*, juntamente com as suas práticas técnicas, arquitetónicas e culturais, representa a convergência de vários movimentos filosóficos e de gestão. Embora muitas organizações tenham desenvolvido estes princípios de forma independente, perceber que o *DevOps* emergiu de uma vasta gama de influências — fenómeno descrito por John Willis como a “convergência do *DevOps*” — revela uma evolução do pensamento, marcada por ligações improváveis. Décadas de ensinamentos provenientes da produção industrial, de organizações de elevada fiabilidade (*high reliability organizations*), de modelos de gestão baseados na confiança (*high-trust*) e de outras abordagens contribuíram para moldar as práticas de *DevOps* tal como as conhecemos hoje (Kim et al., 2016).

1.1 Origens do *DevOps*

O *DevOps* surgiu como uma convergência de várias práticas, filosofias e movimentos de gestão e engenharia. Conforme descrito por John Willis como a “convergência do *DevOps*”, ele incorpora décadas de lições vindas de organizações de alta confiabilidade, produção industrial, engenharia de resiliência, liderança servidora, e culturas de alta confiança. Seu objetivo é melhorar a qualidade, segurança e velocidade na entrega de software em todo o fluxo de valor tecnológico.

1.1.1 Influências

O *DevOps* surgiu como resultado da convergência de diversas correntes filosóficas, técnicas e organizacionais que, ao longo das décadas, influenciaram profundamente a forma como as organizações abordam o desenvolvimento e a operação de software. Entre as principais influências temos (Kim et al., 2016):

- *Lean Manufacturing* e Sistema Toyota de Produção (TPS): Contribuíram com práticas como o mapeamento de fluxo de valor, lotes pequenos, melhoria contínua (*kaizen*) e respeito pelos trabalhadores. O Toyota Kata de Mike Rother destaca o kata de melhoria, uma prática estruturada de aprimoramento diário.

- Teoria das Restrições: Foco em identificar e eliminar gargalos no processo de desenvolvimento e entrega.
- Ágil e o Manifesto Ágil (2001): Trouxe princípios como entrega incremental, feedback contínuo, e equipas pequenas e autónomas, valorizando ciclos curtos de desenvolvimento e interação frequente com o cliente.
- Infraestrutura Ágil e Conferência Velocity (2009): A apresentação “10 Deploys per Day” por Allspaw e Hammond demonstrou a integração entre desenvolvimento e operações, tornando o deployment uma rotina diária. Isso inspirou Patrick Debois a organizar o primeiro DevOpsDays em 2009 na Bélgica, cunhando o termo DevOps.
- Entrega Contínua (*Continuous Delivery*): Formalizada por Jez Humble e David Farley, reforça a importância de um pipeline automatizado e seguro para manter o código em estado implementável constante.

1.1.2 Problemas dos Modelos Tradicionais

O modelo em cascata, dominante antes da era ágil, apresentava algumas desvantagens (Levita, 2017; Scrum Portugal, 2016; Sousa et al., 2019):

- Rigidez na adaptação a mudanças;
- Falta de *feedback* contínuo do cliente;
- Entregas tardias;
- Processos fragmentados entre desenvolvimento e operações;
- Cultura de silos e desconfiança entre equipas.

1.1.3 DevOps como Resposta

DevOps surgiu para superar os conflitos entre equipas de desenvolvimento (focadas em mudanças rápidas) e operações (focadas na estabilidade). Propõe:

- Colaboração transversal entre Dev e Ops (Bass et al., 2015);
- Colaboração transversal e cultura de confiança e responsabilidade partilhada (Kim et al., 2016) ;
- Automatização de processos, desde integração até entrega (Chen, 2015);

- Entrega contínua de valor, com ciclos de feedback curtos (Gruver et al., 2012) ;
- Equipas multidisciplinares com visão e metas comuns (Forsgren et al., 2018).

1.1.4 Definição Moderna de DevOps

O termo *DevOps* surgiu pela primeira vez em 2008. Não há uma única definição universalmente aceite de *DevOps*, já que a interpretação e aplicação do conceito podem variar entre organizações e profissionais. No entanto, uma definição amplamente aceite pela comunidade de *DevOps* é a seguinte:

“DevOps é uma cultura, conjunto de práticas e filosofia de colaboração entre equipas de desenvolvimento (Dev) e operações (Ops) de TI. O principal objetivo do DevOps é automatizar e acelerar a entrega de software, aumentando a eficiência, a qualidade e a confiabilidade das operações de TI.” (Kim et al., 2016)

Essa definição enfatiza a natureza cultural do *DevOps*, destacando a importância da colaboração, da automatização e da entrega contínua de *software*. É importante notar que o *DevOps* não é apenas uma ferramenta ou uma metodologia específica, mas uma abordagem holística que combina práticas técnicas, culturais e organizacionais para melhorar o desenvolvimento e a entrega de *software* (Bass et al., 2015). Embora essa definição seja amplamente reconhecida, a prática do *DevOps* pode variar entre organizações e deve ser adaptada de acordo com as suas necessidades e contextos específicos (Lwakatare et al., 2016). Portanto, é fundamental que as organizações compreendam os princípios e objetivos do *DevOps* e os apliquem de maneira consistente com os seus próprios requisitos e desafios, considerando também os desafios associados, como a resistência cultural e a complexidade da automatização.

1.2 Pilares do DevOps

No ano de 2010, foi estabelecido um modelo conceptual para definir *DevOps*, assente em quatro princípios fundamentais. Esta estrutura ficou conhecida pela sigla CAMS, formada pelas iniciais dos termos ingleses correspondentes (*Culture, Automation, Measurement, Sharing*) (Humble & Molesky, 2011). Num desenvolvimento conceptual posterior, os investigadores incorporaram um quinto elemento - o princípio *Lean* (L) - à estrutura original,

DevOps and (Software) Operational Excellence Metrics

originando assim o acrónimo CALMS. Embora ambas as denominações (a inicial e a ampliada) figurem com frequência em publicações especializadas, a sua prevalência manifesta-se sobretudo em contextos profissionais em detrimento do meio académico, reforçando o carácter aplicado e orientado para a prática que define a filosofia *DevOps*. Segue-se uma exposição detalhada dos cinco componentes fundamentais que constituem o modelo CALMS.

- **Cultura (*Culture*):** A adoção da filosofia *DevOps* introduz um paradigma transformador na cultura empresarial, promovendo uma mudança de mentalidade que fomenta a cooperação interdepartamental e a compreensão mútua entre os diversos atores envolvidos no ciclo de vida do software. Esta abordagem elimina progressivamente as barreiras tradicionais entre departamentos, criando sinergias organizacionais (Humble & Molesky, 2011). O modelo *DevOps* defende a adoção transversal desta mentalidade por todos os departamentos, desde a gestão superior até às equipas técnicas de desenvolvimento e operações. Esta abordagem coloca as pessoas e os processos no centro da estratégia, complementados por ferramentas e métodos que facilitam a entrega sustentada de valor para o cliente final (Sousa, 2019). A cultura *DevOps* valoriza a comunicação eficaz, a confiança entre membros da organização e a autonomia das equipas. A capacidade de tomar decisões locais, experimentar, errar e melhorar progressivamente é incentivada como parte de uma abordagem sistémica à inovação (Walls, 2013).
- **Automatização (*Automation*):** A implementação de fluxos automatizados no desenvolvimento e distribuição de aplicações constitui um pilar fundamental para empresas que ambicionam incorporar práticas *DevOps*. Esta automação abrange múltiplas fases do ciclo de vida do *software* e permite às organizações mecanizar processos e obter diversos benefícios operacionais. Esta abordagem automatizada elimina gargalos tradicionais, criando um fluxo de trabalho contínuo e altamente eficiente que suporta os princípios fundamentais da metodologia *DevOps* (Hamunen, 2016). A automação oferece a vantagem crucial de suprimir atividades rotineiras e mecânicas, transformando-as em ações automáticas, o que evita erros humanos e melhora a consistência da infraestrutura (Lwakatare et al., 2016). Em contrapartida,

DevOps and (Software) Operational Excellence Metrics

otimiza a alocação de talentos especializados em tecnologia, permitindo que esses profissionais se dediquem a atividades estratégicas que gerem maior retorno para a empresa (Trigo et al., 2010). Processos como a Integração Contínua (CI) e a Entrega Contínua (CD) dependem fortemente de pipelines automatizados que garantem que o código, uma vez validado, possa ser colocado em produção de forma segura e previsível. Além disso, a automatização dos testes — tanto unitários como funcionais — permite validar rapidamente a qualidade do software a cada iteração. (Sousa, 2019).

- *Lean*: Inspirado nos princípios do sistema de produção da Toyota, o pensamento *Lean* aplica-se ao desenvolvimento de software como forma de eliminar desperdícios, maximizar a eficiência e gerar valor contínuo para o cliente. Esta filosofia promove ciclos curtos de entrega, feedback constante e melhoria contínua, com base na análise do fluxo de trabalho e na identificação de gargalos (Krafcik, 1988). O DevOps adota esta mentalidade para otimizar todo o ciclo de vida do software, desde o planeamento até à operação em produção. A ênfase recai sobre o aprendizado incremental e a resposta rápida à mudança, fatores essenciais num contexto de elevada incerteza tecnológica e competitividade (Kim et al., 2016).
- *Medição (Measurement)*: A monitorização e análise de dados são fundamentais para avaliar o desempenho das equipas e dos sistemas. No contexto DevOps, a medição deve ser aplicada de forma transversal, abrangendo todas as fases do ciclo de vida do software. Métricas como a frequência de deploys, o tempo médio até à recuperação (MTTR), a taxa de falhas em produção ou o tempo de ciclo são indicadores valiosos da maturidade e eficácia da organização. (Lwakatare et al., 2016). Ao promover uma cultura baseada em dados, as organizações conseguem tomar decisões mais informadas, identificar áreas críticas de melhoria e ajustar processos com base em evidência empírica. Esta visibilidade é crucial para sustentar uma mentalidade de melhoria contínua (Humble & Molesky, 2011).
- *Colaboração (Sharing)*: A partilha de conhecimento, práticas, experiências e até mesmo falhas é incentivada no *DevOps* como forma de reforçar a colaboração e o alinhamento organizacional. Ao contrário de ambientes tradicionais, onde as equipas

funcionam de forma isolada, o *DevOps* promove uma comunicação aberta e transparente, facilitada por plataformas digitais e documentação acessível (Humble & Molesky, 2011). Esta partilha não se restringe ao plano técnico, estendendo-se também a aspetos organizacionais e estratégicos. O objetivo é assegurar que todas as equipas compreendem o impacto do seu trabalho no sistema como um todo, fomentando um sentimento de corresponsabilidade e alinhamento com os objetivos globais da organização (Sousa, 2019). O cerne da transformação *DevOps* está em ultrapassar divisões organizacionais através de soluções tecnológicas que aproximem equipas e conhecimento sistematizado que acelere a produtividade coletiva, particularmente na integração de novos talentos (Walls, 2013).

1.3 Práticas Essenciais do *DevOps*

Com base nos dois modelos apresentados anteriormente, percebe-se que muitos profissionais e investigadores na área de *DevOps* definem esta abordagem através de práticas agrupadas em diferentes categorias. Um dos marcos mais relevantes nesta área é o livro *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*, publicado em 2018 por Nicole Forsgren, Jez Humble e Gene Kim. Esta obra tornou-se uma referência fundamental para as práticas *DevOps*, identificando 24 práticas (ver apêndice I) com impacto na melhoria do desempenho na entrega de software. Estas práticas servem de guia para organizações que pretendem adotar e evoluir as suas práticas *DevOps*.

Ao focarem-se nestas áreas-chave, as organizações conseguem alcançar níveis superiores de desempenho, entregas de software mais rápidas, maior fiabilidade e um alinhamento mais eficaz com as necessidades dos clientes. As práticas foram organizadas em cinco categorias principais: Entrega Contínua, Arquitetura, Produto e Processo, Gestão Lean e Monitorização, e Cultura (Forsgren et al., 2018).

As práticas de Entrega Contínua dizem respeito às capacidades do *DevOps* que permitem construir uma cadeia de entrega contínua para o cliente. Estas práticas centram-se sobretudo na automatização do pipeline *DevOps* e promovem a colaboração entre equipas para atingir um objectivo comum. Além disso, ajudam a proteger o pipeline contra vulnerabilidades e

DevOps and (Software) Operational Excellence Metrics

corrupção de código, tratando aspetos como testes, implementação, entrega, integração e desenvolvimento (Ziaei, 2022).

As práticas de Arquitetura visam a criação de uma linguagem comum e um enquadramento que permitam descrever a organização em termos de negócio. O objectivo é unificar a organização através de fronteiras funcionais, utilizando declarações simples que conectam atividades, resultados e abordagens. No contexto DevOps, isto é possível através da coesão entre equipas e da existência de um pipeline sólido que suporte as suas atividades. O uso de microserviços permite o desenvolvimento, testes e integração contínuos, com forte apoio na automatização de processos e colaboração entre equipas (Ziaei, 2022)..

As práticas de Produto e Processo asseguram que as expectativas dos clientes são cumpridas e que a experiência do utilizador é continuamente melhorada. Estas práticas dependem da qualidade do pipeline DevOps e da competência técnica das equipas. Assim, para evoluir nesta categoria, a organização deve investir na qualificação dos seus profissionais e nas ferramentas adequadas, garantindo a entrega de soluções com qualidade (Ziaei, 2022).

As práticas Culturais dizem respeito aos conhecimentos, competências, comportamentos e sistemas que suportam, planeiam e promovem a entrega de serviços de forma respeitosa e culturalmente adequada. Em DevOps, esta categoria mede a capacidade da equipa de se adaptar e adotar novas práticas e culturas. Para prestar serviços de excelência e construir uma reputação sólida, é fundamental ser flexível perante mudanças — o que implica equipas com uma mentalidade comum e pipelines adaptáveis (Ziaei, 2022).

Por fim, a Gestão Lean representa um modelo de gestão organizacional assente no princípio da melhoria contínua. Esta estratégia de longo prazo visa a otimização progressiva dos processos operacionais, elevando a eficácia e os padrões de qualidade. A gestão lean permite identificar e implementar pequenas mudanças com impacto relevante no pipeline de desenvolvimento. Estas práticas também evidenciam os benefícios de um alinhamento adequado com as condições tecnológicas e ambientais em constante evolução. Para tal, é essencial que a organização apoie as equipas DevOps com recursos e incentivos à mudança positiva (Ziaei, 2022).

DevOps and (Software) Operational Excellence Metrics

O *DevOps* acelera e aprimora o desenvolvimento de software através de métodos ágeis que enfatizam o planeamento flexível e a resposta ágil a mudanças. Esta abordagem cria um fluxo contínuo cliente-desenvolvimento-operações, elevando simultaneamente os padrões qualitativos (Fitzgerald & Stol, 2017). Neste contexto, destaco algumas práticas que considero de importante explicar com maior detalhe:

- **Integração Contínua (CI):** A integração contínua consiste na prática de integrar regularmente alterações de código no repositório partilhado, geralmente várias vezes por dia. Cada submissão é automaticamente testada, o que permite identificar rapidamente erros de integração e assegurar a estabilidade do código principal (trunk). Esta prática reduz significativamente os conflitos de integração e facilita a deteção precoce de defeitos. O CI é frequentemente suportado por pipelines automatizados que executam testes unitários, análises estáticas de código e validações de segurança, assegurando um elevado padrão de qualidade desde as fases iniciais de desenvolvimento (Pequeno, 2022).
- **Entrega Contínua (CD):** Complementar ao CI, a entrega contínua assegura que o software pode ser colocado em produção de forma fiável e segura a qualquer momento. Após a validação automática através de testes, o código é disponibilizado para *deploy* com intervenção mínima, reduzindo o tempo entre desenvolvimento e disponibilização ao utilizador final. Esta prática requer ambientes de teste automatizados, consistência entre ambientes (desenvolvimento, *staging*, produção) e validações finais orientadas por critérios de negócio. A entrega contínua garante maior flexibilidade e capacidade de resposta às mudanças de mercado ou às necessidades dos clientes (Pequeno, 2022).
- **Monitorização Contínua:** A monitorização contínua assegura a visibilidade constante sobre o comportamento dos sistemas em produção. Inclui a recolha de métricas de desempenho, *logs* de aplicação, monitorização de disponibilidade e alertas em tempo real. Esta prática permite detetar falhas precocemente, analisar padrões de utilização e sustentar decisões baseadas em dados reais. A adoção de *dashboards*, painéis de métricas e ferramentas de observabilidade (como *Prometheus*, *Grafana*, *Datadog* ou

DevOps and (Software) Operational Excellence Metrics

Azure Monitor) facilita uma gestão proactiva da infraestrutura e contribui para a resiliência dos serviços digitais. (Pequeno, 2022).

- *Feedback* Contínuo do Cliente: O DevOps institucionaliza um sistema de avaliação permanente com os utilizadores, onde entregas frequentes permitem validações regulares. Esta retroação contínua torna as organizações mais reativas, orienta o desenvolvimento para o essencial e alimenta o planeamento evolutivo - criando um círculo virtuoso de melhoria progressiva (Pequeno, 2022).
- Planeamento Contínuo: A adoção de estratégias de planeamento flexível revela-se crucial para assegurar uma resposta ágil às flutuações do mercado e às sugestões provenientes dos clientes e equipas operacionais. Este enfoque encontra ressonância nos princípios das metodologias ágeis, que preconizam ciclos de planeamento reduzidos e mutáveis, capazes de incorporar ajustes imediatos quando necessário (Virmani, 2015).

A metodologia *DevOps* potencia esta abordagem através da manutenção de um inventário de funcionalidades (*product backlog*), cuja hierarquização é continuamente reavaliada com base no input dos utilizadores. Este mecanismo assegura que a priorização das tarefas se mantém alinhada com as necessidades reais dos clientes (Virmani, 2015).

1.4 Ciclo do *DevOps*

O fluxo de trabalho DevOps, ilustrado na Figura 1.1, organiza-se em etapas interligadas que promovem a colaboração entre desenvolvedores e administradores, garantindo uma entrega contínua e eficiente de código em produção. Representado frequentemente como um *loop* infinito, o ciclo inclui: planeamento, desenvolvimento, integração e testes, entrega, operação e monitorização (Pequeno, 2022; Trigo et al., 2022).

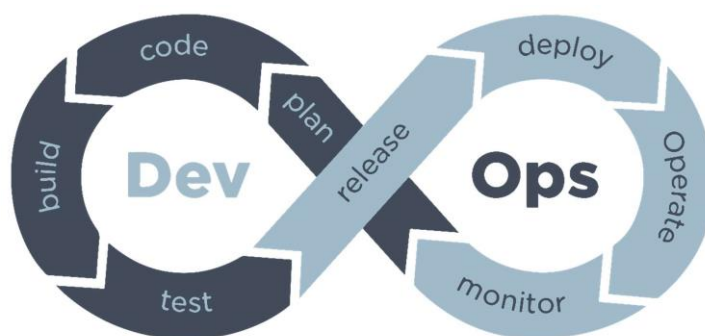


Figura 1.1. Pipeline de DevOps

Fonte: Sousa (2019)

O ciclo começa com a definição de objetivos e requisitos alinhados às necessidades do negócio, envolvendo *stakeholders* e equipas técnicas desde o início. Utilizam-se metodologias ágeis como *Scrum* ou *Kanban*, com foco em *backlogs* e *sprints*. Na fase de *build*, ocorrem a compilação e os testes unitários; na de testes, são aplicadas validações funcionais e de qualidade. Ferramentas de versionamento, revisão de código e análise estática asseguram qualidade desde cedo. A integração contínua automatiza testes e acelera a deteção de falhas. As fases de *release* e *deploy* envolvem a disponibilização e instalação do código, respetivamente. Técnicas como infraestrutura como código e containers reduzem riscos e agilizam o processo. Já a fase operacional foca-se na monitorização, estabilidade e desempenho, com deteção proativa de anomalias, correções, escalabilidade e segurança (Sousa, 2019; Sousa et al., 2019). A recolha e partilha de métricas operacionais alimenta o ciclo com dados, guiando decisões e melhorias (Amaro et al., 2024; Ziaei, 2022). Assim, o DevOps é um processo contínuo de evolução, sustentado pela integração de pessoas, processos e tecnologias, que garante a entrega de valor constante ao cliente.

1.5 Métricas de DevOps

Uma avaliação de desempenho eficaz em contextos DevOps deve assentar em dois princípios fundamentais: a valorização dos resultados coletivos, promovendo a colaboração entre equipas, e a avaliação com base no impacto real do trabalho, em vez de simplesmente medir a quantidade de produção. Recompensar *developers* pelo volume de entregas e as equipas de operações pela estabilidade pode gerar conflitos de interesse, comprometer a

DevOps and (Software) Operational Excellence Metrics

qualidade do código e originar processos burocráticos e ineficientes. É, por isso, essencial alinhar os incentivos com os objetivos estratégicos da organização, evitando premiar esforços que não produzem resultados relevantes (Forsgren et al., 2018). Não deve existir um compromisso entre desempenho e qualidade: um pipeline de integração e entrega contínuas (CI/CD) bem estruturado deve permitir melhorias simultâneas em velocidade, estabilidade e eficiência. A monitorização de métricas como a velocidade das atividades, a qualidade do software e o nível de automatização do pipeline permite identificar áreas de melhoria e verificar o impacto das alterações implementadas. As métricas de DevOps, que avaliam tanto o funcionamento do pipeline como o desempenho das equipas, fornecem dados regulares que permitem acompanhar tendências e propor ajustamentos sempre que necessário (Ziaei, 2022). A análise contínua destas métricas é fundamental para garantir entregas de software eficazes, alinhadas com os objetivos estratégicos da organização. Estas métricas permitem ainda avaliar o progresso em direção à excelência técnica, à satisfação do cliente e à manutenção de uma vantagem competitiva, sendo escolhidas de acordo com as metas específicas de cada organização (Ziaei, 2022).

As métricas de *DevOps* são utilizadas para planeamento, monitorização de processos, avaliação e melhoria da qualidade, identificação e resolução de problemas, bem como para motivação das equipas. Constituem ferramentas de medição que fornecem informações cruciais sobre (Kurkela, 2020):

- A eficácia das práticas *DevOps*;
- A qualidade do software entregue;
- A eficiência dos processos;
- O impacto nos resultados organizacionais.

Estas métricas permitem que equipas *DevOps*, gestores e *stakeholders* acompanhem o progresso, identifiquem áreas de melhoria e tomem decisões assentes em dados concretos. É importante salientar que a seleção e aplicação adequadas das métricas são essenciais para obter resultados significativos, evitando erros comuns como a utilização de métricas mal concebidas, que podem gerar comportamentos contraproducentes (Kurkela, 2020).

1.5.1 Métricas DORA

As métricas DORA (*DevOps Research and Assessment*) constituem um conjunto de indicadores concebidos para avaliar o desempenho de equipas de engenharia de software em ambientes ágeis e *DevOps*. Desenvolvidas a partir da investigação da DORA — organização fundada em 2014 pelos especialistas Nicole Forsgren, Jez Humble e Gene Kim —, estas métricas visam identificar as práticas mais eficazes e os indicadores-chave para a melhoria contínua do desempenho das equipas.

O estudo anual "*Accelerate State of DevOps Report*", baseado nas pesquisas da DORA, tornou-se uma referência global no setor, oferecendo insights valiosos para a otimização de processos de desenvolvimento de *software* e operações. O relatório baseia-se em dados de pesquisas com milhares de profissionais de *software* e TI, analisando estatisticamente os dados para identificar práticas eficazes. O objetivo principal é identificar as práticas e métricas que distinguem as organizações de alto desempenho das de baixo desempenho, com foco em fatores como frequência de implementação, taxas de falha, ferramentas *DevOps* e práticas de segurança. A metodologia DORA assenta em quatro métricas principais, agrupadas em duas dimensões (DORA, 2022):

- Velocidade (tempo de implementação e frequência de lançamento);
- Estabilidade (tempo de restauro e taxa de falhas).

Esta abordagem permite uma avaliação equilibrada do desempenho das equipas.

O tempo de execução (*lead time*) refere-se ao período entre o pedido do cliente e a sua satisfação. No contexto de desenvolvimento de produtos - onde se pretende satisfazer múltiplos clientes de formas não necessariamente antecipadas - este tempo divide-se em dois componentes distintos (Forsgren et al., 2018):

1. Tempo de conceção e validação - período necessário para projetar e validar um produto ou funcionalidade;
2. Tempo de entrega - período necessário para implementar, testar e disponibilizar a funcionalidade aos clientes.

Na fase de conceção, a medição do tempo apresenta desafios significativos:

DevOps and (Software) Operational Excellence Metrics

- Ponto de início frequentemente indefinido;
- Elevada variabilidade temporal.

Já na fase de entrega:

- A medição é mais objetiva;
- A variabilidade é consideravelmente menor.

A tabela seguinte ilustra esta distinção entre os dois domínios temporais.

Tabela 1.1. *Design vs. Entrega*

Design e Desenvolvimento de Produtos	Entrega do Produto (<i>Build, Testing, Deployment</i>)
Criar novos produtos e serviços que resolvam problemas dos clientes utilizando uma abordagem de entrega baseada em hipóteses, uma experiência do utilizador moderna e <i>design thinking</i> .	Possibilitar um fluxo rápido desde o desenvolvimento até à produção e lançamentos fiáveis através da padronização do trabalho e da redução do tamanho dos <i>batches</i> e da variabilidade.
O <i>design</i> e implementação de funcionalidades podem exigir trabalho que nunca foi realizado anteriormente.	A integração, teste e implementação devem ser realizados de forma contínua e o mais rapidamente possível.
As estimativas são altamente incertas.	Os tempos de ciclo devem ser conhecidos e previsíveis.
Os resultados são altamente variáveis.	Os resultados devem apresentar baixa variabilidade.

Fonte: Forsgren et al. (2018)

Prazos de entrega mais curtos são melhores, pois permitem obter um *feedback* mais rápido sobre o que estamos a construir e ajustar o rumo de forma mais ágil. Além disso, prazos curtos são importantes quando há um defeito ou interrupção, pois possibilitam fornecer uma correção rápida e com elevada confiança. Segundo os autores, o *Lead Time* (prazo de entrega do produto) é medido como o tempo que decorre desde o *commit* do código até este estar a ser executado com sucesso em produção (Forsgren et al., 2018). Reduzir o tamanho do *batch* é outro elemento central do paradigma *Lean* — de facto, foi um dos pilares do sucesso do sistema de produção da Toyota. A diminuição do tamanho dos *batches* reduz os tempos de ciclo e a variabilidade no fluxo, acelera o *feedback*, diminui o risco e os custos indiretos, aumenta a eficiência, reforça a motivação e o sentido de urgência, e reduz tanto os

DevOps and (Software) Operational Excellence Metrics

custos como a expansão do cronograma. No entanto, em software, o tamanho do *batch* é difícil de medir e comunicar entre diferentes contextos, já que não existe um inventário visível. Por isso, os autores optaram por usar a frequência de *deployment* como um *proxy* para o tamanho do *batch*, dado que é fácil de medir e tende a ter baixa variabilidade. Por *deployment*, entende-se a implementação do software em produção ou numa *app store* (Forsgren et al., 2018). Os tempos de entrega e a frequência de implementação são ambos indicadores-chave de desempenho na entrega de *software*. Tradicionalmente, a fiabilidade media-se pelo tempo entre falhas. No entanto, em sistemas de *software* modernos — que são sistemas complexos em constante evolução — as falhas são inevitáveis. Assim, a questão fundamental passa a ser: Quão rapidamente se consegue recuperar o serviço (Forsgren et al., 2018). Por fim, uma métrica fundamental ao implementar alterações em sistemas é a taxa de falhas nas mudanças em produção - incluindo, por exemplo, lançamentos de *software* e alterações de configuração de infraestrutura (Forsgren et al., 2018).

Segundo o *Accelerate State of DevOps Report (2022)*, as quatro métricas fundamentais de desempenho na entrega de *software* podem ser consideradas em termos de *throughput* e estabilidade. O *throughput* é medido através do tempo de execução para alterações de código (ou seja, o tempo desde a confirmação do código até ao lançamento na produção) e a frequência de implementação. A estabilidade é medida recorrendo ao tempo para restabelecer um serviço após um incidente e à taxa de falhas nas alterações. Com oito anos de investigação e mais de 33.000 respostas a inquéritos de profissionais do setor, o *State of DevOps Report* identifica as práticas de desenvolvimento de *software* e *DevOps* que potenciam o sucesso de equipas e organizações. Em 2022, mais de 1350 profissionais ativos de diversos setores e países partilharam as suas experiências, contribuindo para alargar o conhecimento sobre os fatores críticos para um desempenho de excelência. A tabela seguinte classifica cada uma das quatro métricas DORA em três categorias (DORA, 2022).

DevOps and (Software) Operational Excellence Metrics

Tabela 1.2. Software Delivery Performance

Métrica de Desempenho na Entrega de Software	Baixa	Média	Alta
<i>Deployment frequency.</i> Qual é a frequência com que a sua organização coloca código em produção ou o disponibiliza aos utilizadores finais?	Menos de uma vez a cada 6 vezes.	Entre uma vez por mês e uma vez a cada 6 meses	<i>On-demand</i> (várias implementações por dia)
<i>Lead time for changes.</i> Qual é o lead time das alterações (ou seja, quanto tempo decorre desde o <i>commit</i> do código até à sua execução bem-sucedida em produção)?	Entre um a seis meses	Entre uma semana e um mês	Entre um dia e uma semana
<i>Time to restore service.</i> Qual é o tempo médio de recuperação (MTTR) após uma falha no serviço que impacte os utilizadores (ex.: indisponibilidade não planeada ou degradação significativa do desempenho)?	Entre uma semana e um mês	Entre um dia e uma semana	Menos de um dia
<i>Change failure rate.</i> Qual é a percentagem de alterações implementadas na produção ou lançadas para os utilizadores que resultam num serviço degradado (por exemplo, levando a uma deterioração ou interrupção do serviço) e subsequentemente requerem correção (por exemplo, necessitam de um <i>hotfix</i> , <i>rollback</i> , <i>fix forward</i> , ou <i>patch</i>)?	46%-60%	16%-30%	0%-15%

Fonte: DORA (2022)

O relatório *Accelerate State of DevOps* (2024) assinalou a 10.^a edição do estudo DORA, sintetizando uma década de análises sobre práticas de alto desempenho em organizações tecnológicas. Baseado em dados de mais de 39 mil profissionais a nível global, o relatório deste ano incluiu contribuições de cerca de 3 mil especialistas de diversos setores e nacionalidades (DORA, 2024).

Neste ano, a equipa de investigação DORA refinou significativamente a abordagem para medição de tempos de recuperação, implementando uma distinção crucial na metodologia de avaliação. Anteriormente, a questão genérica - "Para a aplicação ou serviço principal em que trabalha, quanto tempo demora geralmente a restaurar o serviço quando ocorre um incidente ou defeito que impacta os utilizadores?" - apresentava limitações conceptuais importantes, pois agregava indiscriminadamente (DORA, 2024):

DevOps and (Software) Operational Excellence Metrics

1. Falhas originadas por alterações de *software* (implementações problemáticas);
2. Interrupções causadas por fatores externos (ex.: catástrofes naturais, falhas de infraestrutura).

Esta agregação impedia uma análise precisa do desempenho específico do ciclo de entrega de *software*, misturando variáveis não controláveis com indicadores diretamente relacionados com processos de desenvolvimento e operações.

Para superar esta limitação, introduziram uma reformulação metodológica com três componentes essenciais (DORA, 2024):

1. Precisão Conceptual: Adotaram a métrica específica "Tempo de Recuperação de Implementações com Falha" (*Failed Deployment Recovery Time*), que se foca exclusivamente em incidentes desencadeados por:
 - Alterações de código em produção;
 - Lançamentos de novas versões;
 - Atualizações de configuração.
2. Diferenciação Terminológica: Esta abordagem permite distinguir claramente entre:
 - MTTR genérico (*Mean Time to Restore*): Medida abrangente de todos os incidentes;
 - FDRT (*Failed Deployment Recovery Time*): Indicador específico para falhas de implementação.
3. Alinhamento Estatístico: A nova formulação - "Tempo para restaurar o serviço após uma alteração em produção resultar em degradação do serviço e necessitar de remediação (*hotfix, rollback* ou *patch*)" - apresenta vantagens metodológicas:
 - Elimina a ambiguidade inerente ao MTTR (questões sobre média vs. mediana);
 - Permite comparações válidas entre equipas e organizações;
 - Alinha-se com os princípios da engenharia de resiliência moderna;
 - Facilita a identificação de padrões em falhas relacionadas com o ciclo de entrega.

Esta evolução reflete a maturação das práticas de medição no domínio *DevOps*, onde:

- Se reconhece a inadequação progressiva do MTTR como métrica única;

DevOps and (Software) Operational Excellence Metrics

- Se valoriza a contextualização das análises de falhas;
- Se privilegiam indicadores específicos para domínios específicos.

A implementação desta métrica refinada permite às equipas:

- Identificar com maior precisão pontos de melhoria nos processos de implementação;
- Isolar variáveis externas na avaliação de desempenho;
- Direcionar esforços de otimização para áreas com maior potencial de retorno.

Esta abordagem está totalmente alinhada com o quadro conceptual, que enfatiza a necessidade de métricas específicas e acionáveis para a melhoria contínua do desempenho de entrega de *software*. A tabela seguinte mostra como os participantes do estudo de 2024 se distribuem em termos de desempenho na entrega de *software* (DORA, 2024).

Tabela 1.3. Software Delivery Performance

Performance level	Change lead time	Deployment frequency	Change fail rate	Failed deployment recovery time	Percentage of respondents*
Elite	Menos de um dia	<i>On demand</i> (várias implementações por dia)	5%	Menos de uma hora	19% (18-20%)
High	Entre um dia e uma semana	Entre uma vez por dia e uma vez por semana	20%	Menos de um dia	22% (21-23%)
Medium	Entre uma semana e um mês	Entre uma vez por semana e uma vez por mês	10%	Menos de um dia	35% (33-36%)
Low	Entre um mês e seis meses	Entre uma vez por mês e uma vez a cada 6 meses	40%	Entre uma semana e um mês	25% (23-26%)

* 89% intervalo de incerteza

Fonte: DORA (2024)

As métricas DevOps são instrumentos fundamentais para avaliar o desempenho do pipeline e das equipas, permitindo uma monitorização contínua da produtividade e facilitando a identificação de tendências, melhorias ou deteriorações ao longo do tempo produtividade (Perera et al., 2016; Roche, 2019). Estas métricas fornecem dados que orientam alterações

estratégicas, contribuindo para a evolução contínua da equipa. Ao medir tanto as capacidades técnicas como os processos, estas métricas oferecem uma visão abrangente da eficiência da equipa, sendo essenciais para determinar o seu valor comercial e o seu grau de maturidade DevOps. Indicadores como o tempo de implementação de alterações e a taxa de falhas permitem avaliar a capacidade de resposta às exigências do mercado, promovendo a correção de falhas e a melhoria contínua da qualidade (Katal et al., 2019). Assim, estas métricas não só ajudam a reduzir a distância entre desenvolvimento e operações, como reforçam a colaboração entre programadores e administradores de sistemas (Shah & Dubaria, 2019). São também essenciais para acompanhar o progresso rumo a objetivos estratégicos, como a aceleração dos ciclos de lançamento, a manutenção da vantagem competitiva e a satisfação do cliente. A sua escolha depende dos objetivos específicos da organização, sendo que um conjunto de dez métricas principais é geralmente considerado suficiente para fornecer uma visão clara do desempenho da equipa e orientar ações de melhoria (Ziaei, 2022).

1.5.2 Métricas de DevOps selecionadas

Nas secções seguintes apresentam-se as métricas de DevOps mais importantes encontradas na literatura.

1.5.2.1 Lead Time for Change

O *lead time for change* (tempo de implementação de alterações) também pode ser referido simplesmente como *lead time*. É definido como "o tempo necessário desde a submissão do código até ao seu funcionamento efetivo em produção". Um *lead time* mais curto oferece vantagens como obtenção de *feedback* mais rápido, capacidade de correção ágil e entrega mais veloz de soluções para problemas. Trata-se de um indicador fundamental para qualquer equipa *DevOps* (Ziaei, 2022). O *lead time* representa a duração total necessária para completar uma tarefa específica. Em termos práticos, corresponde ao período entre o início de uma tarefa e o momento da sua implementação. Este parâmetro reflete o nível de desempenho da equipa *DevOps*. Por exemplo, equipas de alto desempenho medem o *lead time* em horas. Contudo, esta métrica pode não se aplicar a todas as tarefas atribuídas a uma equipa *DevOps*. Em alguns casos, as tarefas podem ser mais complexas e exigir mais tempo. A melhor forma de otimizar o *lead time* é trabalhar com pequenos lotes de trabalho

(Maroukian & Gulliver, 2020). Desta forma, a medição do *lead time* permite que os desenvolvedores obtenham feedback rápido e possam implementar as alterações necessárias ou manter o desenvolvimento em curso. De acordo com o estudo, 9 em cada 16 artigos analisados definem o *change lead time* (Ziaei, 2022). Todas as recomendações baseadas na definição original quantificam o tempo que um *commit* leva até chegar à produção, ou seja, o *change lead time* é determinado pelo intervalo de tempo entre o *commit* e a implementação.

1.5.2.2 Change Failure Rate

A eficácia do *software* pode determinar rapidamente o sucesso das equipas *DevOps* na implementação de alterações com probabilidade reduzida ou nula de falha. Define-se como a percentagem de alterações numa aplicação ou serviço que resultam em degradação do serviço ou necessitam de intervenção corretiva (como falhas que causam indisponibilidade do serviço, exigindo reparação de emergência, *rollback*, correções diretas ou aplicação de *patches*). As equipas consideradas de alto desempenho mantêm taxas de falha em alterações entre 0-15% (Callanan & Spillane, 2016). Esta métrica permite ainda que as falhas no processo sejam identificadas e resolvidas atempadamente. Uma equipa *DevOps* com baixa probabilidade de taxa de falha em alterações distingue-se pela implementação de testes automatizados, desenvolvimento baseado no tronco principal (*trunk-based development*) e trabalho em pequenos incrementos (Wiedemann et al., 2019). Deste modo, a monitorização contínua da taxa de falha nas alterações revela-se essencial para detetar e resolver eventuais problemas. Segundo o estudo, 9 em cada 16 artigos académicos incluem uma definição para taxa de falha em alterações (Ziaei, 2022). Apresentam-se de seguida metodologias alternativas para a sua medição:

- Percentagem de *releases* que obrigaram a um "lançamento de correção";
- Número de *hotfixes* referenciados nas mensagens de *commit*;
- Identificação de falhas através da análise de dados de monitorização organizados por *deployments*;
- Classificação explícita de cada *deployment* como bem-sucedido ou falhado;
- Cálculo do rácio entre o número de *rollbacks* e o total de *deployments* realizados.

Para apurar a taxa de falha, importa primeiro estabelecer o que constitui uma alteração. Na literatura especializada, cada *deployment* é considerado uma alteração. Assim, a taxa de falha calcula-se dividindo o número de alterações problemáticas pelo total de alterações implementadas. Subsequentemente, é necessário caracterizar o que configura uma falha e estabelecer a sua relação causal com a alteração correspondente (Ziaei, 2022).

1.5.2.3 Deployment Frequency

Tal como qualquer outra métrica, a frequência de implementação (*deployment frequency*) constitui um indicador crucial para avaliar o desempenho de uma equipa *DevOps*. Esta métrica encontra-se diretamente relacionada com a redução do volume dos lotes de trabalho. A periodicidade com que o *software* é implementado em produção serve como parâmetro de avaliação, sendo que uma frequência de implementação adequada permite efetuar alterações *on-demand* e com regularidade (Haindl & Plösch, 2019). A estratégia de implementação adotada pela equipa *DevOps* revela-se determinante para o seu desempenho global. A eficácia dos sistemas automatizados no processo de implementação, sublinhando que os testes automatizados e mecanismos de feedback reduzem significativamente a necessidade de intervenção manual (Erich et al., 2017). Assim sendo, ciclos de implementação rápidos refletem um elevado nível de desempenho da equipa *DevOps*, enquanto atrasos ou uma frequência de implementação reduzida apontam para capacidades mais limitadas (Trihinas et al., 2018). De acordo com um estudo recente, 7 em cada 16 artigos científicos analisados apresentam uma definição explícita para *deployment frequency* (Ziaei, 2022). As diversas definições não divergem significativamente, uma vez que este conceito se encontra claramente estabelecido como "a implementação de software em ambiente de produção". Em todos os casos analisados, é contabilizado o "número de implementações/lançamentos efetuados num determinado intervalo temporal". Alguns artigos referem especificamente que monitorizam apenas implementações bem-sucedidas (embora não definam explicitamente os critérios de sucesso), enquanto outros incluem no todas as implementações em produção.

1.5.2.4 Mean Time to Recovery

Normalmente, os sistemas estão sujeitos a falhas sistemáticas ou operacionais. A capacidade de uma equipa recuperar dessas falhas e retomar o normal funcionamento é quantificada através do tempo médio de recuperação (MTTR - *Mean Time To Recovery*). Este conceito parte do princípio de que, em sistemas complexos de rápida evolução, as falhas são inevitáveis - sendo crucial determinar quanto tempo demora a restauração do serviço após um incidente (como uma paragem inesperada ou degradação de serviço), medido desde o momento em que o problema ocorre (Ziaei, 2022). As equipas *DevOps* de alto desempenho destacam-se pela capacidade de recuperação rápida de falhas do sistema. O principal objetivo desta métrica é duplo: reduzir a frequência de falhas e garantir que, quando ocorrem, possam ser resolvidas eficientemente. A monitorização contínua, através de ferramentas robustas de observabilidade de aplicações, revela-se essencial para detetar anomalias do sistema atempadamente, permitindo uma resposta mais ágil. Esta métrica, expressa em horas, serve como termómetro do desempenho da equipa *DevOps* (Ziaei, 2022). Para otimizar o tempo médio de recuperação, é fundamental que a equipa operacional receba notificações imediatas em caso de falha do sistema. Paralelamente, a equipa deve dispor dos métodos adequados, ferramentas especializadas e privilégios necessários para resolver problemas sistémicos (Ziaei, 2022). O MTTR constitui assim um indicador fiável para avaliar a velocidade, qualidade e eficácia global de uma equipa *DevOps*. No que concerne à definição e cálculo do MTTR, um estudo revela que 8 em cada 16 artigos académicos abordam o tempo de restauro de serviço. Cinco destes definem-no como o tempo médio necessário para resolver um incidente dentro de um período específico. Entre as metodologias propostas, destaca-se a engenharia do caos (que envolve a simulação intencional de falhas para medir os tempos de deteção e resolução), bem como a verificação periódica do estado do sistema, registando-se o tempo necessário para corrigir eventuais degradações detetadas (Ziaei, 2022). A literatura especializada converge na definição de tempo de restauro como o período compreendido entre o início de um incidente e a sua efetiva resolução.

1.5.2.5 Customer Ticket

A satisfação do cliente representa provavelmente o objetivo mais crítico de qualquer departamento organizacional. A métrica de *tickets* de cliente funciona como um indicador valioso que fornece aos *developers* do sistema *feedback* sobre a sua operacionalidade - seja em termos de fluidez ou complexidade. Embora a equipa *DevOps* implemente diversos controlos, alguns *bugs* e falhas ocasionais podem passar inadvertidos. A recorrência com que a equipa recebe tickets de cliente reportando erros ou problemas técnicos constitui um espelho fiel do seu desempenho (John et al., 2017). A quantidade de *tickets* que sinalizam questões de robustez ou qualidade do sistema serve como termómetro claro do desempenho da equipa *DevOps*. É essencial que a equipa possua capacidade e recursos para responder eficazmente às solicitações dos clientes, assegurando que os *tickets* recebidos refletem genuinamente a resiliência do sistema. Por lidar diretamente com a experiência do utilizador final, a métrica de *tickets* de cliente oferece simultaneamente a melhor e mais desafiante oportunidade para avaliar o desempenho de uma equipa *DevOps* (John et al., 2017).

1.5.2.6 Defect Escape Rate

O número de defeitos identificados no software entre o desenvolvimento e a implementação constitui a taxa de defeitos. A capacidade de resolver esses defeitos de forma oportuna enquadra-se na métrica de taxa de fuga de defeitos. Várias iniciativas podem ser adotadas para detetar estes problemas durante a fase de desenvolvimento no *pipeline*, salientando a importância de testes contínuos e da monitorização rigorosa do processo de desenvolvimento (Ziaei, 2022). A quantidade de defeitos que atingem a fase de produção reflete diretamente o nível de desempenho da equipa *DevOps*. Deste modo, uma equipa que identifica e corrige defeitos nas etapas iniciais do ciclo de desenvolvimento evidencia um elevado nível de eficiência. Convém notar que, ocasionalmente, alguns destes defeitos podem escapar aos processos de teste e monitorização. No entanto, a sua identificação prévia à fase de controlo de qualidade e aos departamentos de produção revela-se crucial (Ziaei, 2022).

1.5.2.7 Automated Test Percentage

A velocidade e o alto desempenho figuram entre os principais objetivos das métricas *DevOps*. Para atingir estes objetivos, a equipa *DevOps* deve incorporar e utilizar tecnologias que permitam monitorizar e testar integralmente o pipeline de produção (Maroukian & Gulliver, 2020). Esta métrica foca-se particularmente na eficácia dos testes automatizados. Paralelamente, a monitorização da duração dos testes automatizados face à frequência de alterações de código constitui um parâmetro fundamental para qualquer equipa *DevOps* (Gunja, 2021). A rapidez na identificação de *bugs* e erros no sistema deve ser uma prioridade absoluta para toda a equipa. As equipas de excelência destacam-se precisamente pelo acompanhamento rigoroso das taxas de cobertura dos testes automatizados (Gunja, 2021).

1.5.2.8 Deployment Size

O número total de implementações representa outra métrica crucial que a equipa *DevOps* deve acompanhar. Embora existam outros indicadores, como a frequência de implementação e as implementações malsucedidas, a dimensão das implementações permite à equipa avaliar o volume de trabalho que consegue processar num determinado intervalo temporal (Torble, 2019). Esta métrica permite ainda identificar a tipologia e o volume diário de implementações realizadas pela equipa. A eficiência da equipa *DevOps* pode ser aferida através da análise sistemática das métricas de dimensão de implementação (Singh, 2024). Importa referir que a dimensão das implementações está condicionada por variáveis como a frequência, a complexidade e a quantidade de implementações.

1.5.2.9 Error Rates

Os sistemas de *software* e aplicações estão naturalmente propensos a erros. Consequentemente, o acompanhamento das taxas de erro revela-se essencial ao longo de todo o ciclo de desenvolvimento. A ocorrência de erros desde a fase de desenvolvimento até à implementação deve ser minimizada para espelhar o elevado desempenho da equipa (Ziaei, 2022). Paralelamente, a resolução atempada de quaisquer erros detetados no *pipeline* representa uma abordagem estratégica eficiente no tratamento de *bugs* e incidentes de produção. A adoção de ferramentas automatizadas de teste e monitorização assume

particular relevância na deteção precoce de anomalias. Importa reconhecer que qualquer sistema conterà inevitavelmente alguns erros, dada a intrínseca complexidade dos ambientes tecnológicos (Ziaei, 2022). Todavia, compete à equipa *DevOps* assegurar a identificação e resolução da esmagadora maioria desses erros.

1.5.2.10 Mean Time to Detection

O tempo médio até à deteção (*Mean Time to Detection* - MTTD) representa um parâmetro fundamental que avalia a capacidade de identificação de falhas no sistema (Ziaei, 2022). A ocorrência de falhas sistémicas ou interrupções nas fases finais de desenvolvimento não pode ser encarada como situação normal em qualquer projeto. Esta métrica permite avaliar a eficiência da equipa na deteção imediata de erros e anomalias após o seu surgimento. Uma equipa *DevOps* competente caracteriza-se por um tempo médio de deteção extremamente reduzido. À semelhança de outras métricas, a implementação de sistemas avançados de monitorização e testes automatizados permite otimizar significativamente este indicador temporal (Gunja, 2021).

1.6 Utilização das métricas

A investigação científica na área de *DevOps* sublinha que, apesar de existir um conjunto amplamente reconhecido de métricas, a sua aplicação eficaz está longe de ser universal.

Pelo contrário, a seleção e interpretação destas métricas devem ser cuidadosamente adaptadas ao contexto específico de cada organização. Embora estas métricas sirvam como referências valiosas, a sua relevância varia consoante fatores como (Forsgren et al., 2018):

- A maturidade *DevOps* da organização (empresas em fase inicial podem priorizar métricas de cultura e colaboração, enquanto equipas mais avançadas focam-se em eficiência operacional);
- O sector de atividade (ex.: *startups* de *software* podem valorizar *deployment frequency*, enquanto empresas financeiras dão primazia a *stability* e *security compliance*);
- A complexidade dos sistemas (sistemas *legacy* exigem métricas diferentes de ambientes *cloud-native*);

DevOps and (Software) Operational Excellence Metrics

- Os objetivos estratégicos (melhorar a velocidade de entrega vs. aumentar a fiabilidade).

Além disso, a literatura alerta para o risco de adotar métricas de forma rígida, sem considerar a cultura interna ou os processos únicos de cada equipa. Por exemplo, uma métrica como *Deployment Frequency* pode ser irrelevante para uma equipa que trabalha com *embedded systems*, onde os lançamentos são menos frequentes por natureza. Do mesmo modo, o MTTR pode ter interpretações distintas consoante a criticidade do serviço afetado.

Assim, conclui-se que o verdadeiro valor das métricas *DevOps* reside não na sua aplicação genérica, mas na capacidade de as ajustar dinamicamente, garantindo que refletem as prioridades e realidades específicas de cada organização. Esta abordagem contextualizada aliada a uma avaliação contínua é essencial para evitar conclusões enganadoras ou incentivos distorcidos. A flexibilidade na seleção e interpretação das métricas constitui, portanto, um elemento fundamental para evitar análises distorcidas ou incentivos contraproducentes. Conclui-se que, em vez de adotar métricas universais de forma indiscriminada, as organizações devem ajustá-las continuamente, alinhando-as com os seus processos e objetivos específicos, de modo a obter informações verdadeiramente úteis e acionáveis.

2 METODOLOGIA

Tendo em conta a natureza deste trabalho, a metodologia *Design Science Research* (DSR) foi a selecionada para alcançar o objetivo do estudo. A *Design Science Research Methodology* (DSRM) define-se como uma abordagem que visa desenvolver e estudar funcionalmente um artefacto ou instrução (Romme, 2003), sendo considerada um processo abrangente orientado para a criação de artefactos com o propósito de resolver problemas específicos (March & Storey, 2008). Além disso, a DSR é reconhecida como uma abordagem inovadora que possibilita o desenvolvimento de investigação em diversas áreas, podendo ser utilizada como ferramenta para produzir conhecimento tanto académico como organizacional (Romme, 2003). A aplicação da DSR contribui para colmatar o fosso entre a teoria e a prática, não apenas permitindo resolver problemas organizacionais e académicos, mas também gerando conhecimento com potencial para aprimorar as teorias existentes (Romme, 2003).

Os princípios da DSR assentam na engenharia de sistemas artificiais, sendo os sistemas de informação (SI) um exemplo paradigmático de sistemas artificiais quando o objetivo é potenciar a eficiência organizacional. Assim, na Tabela 2.1, é possível verificar a forma como os princípios foram adotados.

Tabela 2.1. Design science research principles

Princípio DSR	Explicação
Abstração	O estudo envolve o desenvolvimento de um <i>dashboard</i> para monitorização de dados em ambientes <i>DevOps</i> , visando apoiar os <i>stakeholders</i> no processo decisório. Seguiram-se as melhores práticas do setor e analisaram-se metodologias consolidadas para validação do conceito.
Originalidade	Embora os <i>dashboards</i> sejam comuns em <i>DevOps</i> , esta solução oferece uma abordagem integrada e personalizável, distinguindo-se das existentes no mercado.
Fundamentação	O <i>dashboard</i> foi desenvolvido com base numa metodologia robusta, validada posteriormente por especialistas do setor, assegurando a sua pertinência prática.
Benefício	A solução consolida múltiplas vertentes <i>DevOps</i> num único interface, permitindo aos <i>stakeholders</i> decisões mais ágeis e fundamentadas.

Fonte: Elaboração própria com adaptação de Hevner et al. (2004)

Tendo como objetivo principal o desenvolvimento de um *dashboard* em *Power BI* para o grupo ANOVA, que centraliza e automatiza o *pipeline* de produção de métricas *DevOps* necessárias à monitorização e alcance dos objetivos operacionais no contexto de desenvolvimento e operação de *software*, a metodologia adotada foi a *Design Science Research (DSR)*.

Como o nome sugere, esta metodologia consiste numa “ciência do *design*”, baseada na criação de um artefacto para resolver o problema em causa. A DSR desenvolve novas práticas e técnicas para melhorar tanto a eficácia como a eficiência na resolução do problema. O mais importante é que a investigação e o desenvolvimento sejam relevantes para solucionar uma questão de negócio ainda não resolvida (Peppers et al., 2007).

Assim, o desenvolvimento do artefacto deve ser um processo de investigação fundamentado em teorias e conhecimentos existentes, de modo a encontrar uma solução para o problema definido. A investigação em ciência do design tem como propósito desenvolver artefactos que materializem a visão ideal de um sistema (Geerts, 2011), promovendo a transição do estado atual para o estado pretendido.

Na Figura 2.1. está esquematizado a metodologia DSR.

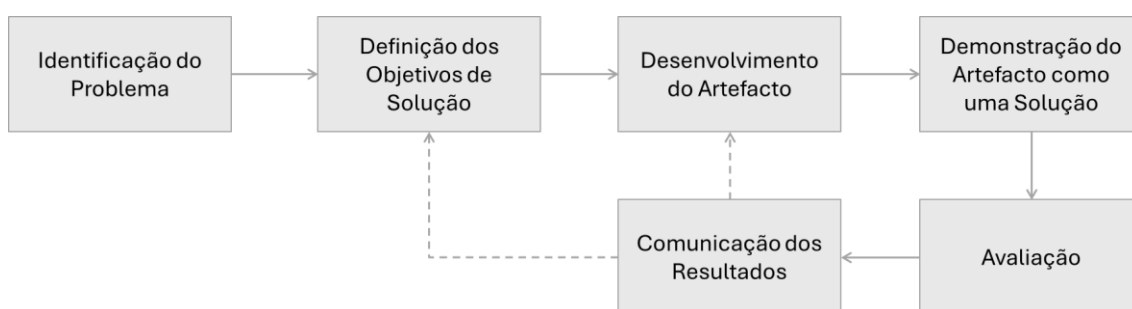


Figura 2.1. Metodologia DSR

Fonte: Elaboração própria com adaptação de Peppers et al. (2007)

A primeira fase da metodologia DSR foca-se na identificação do problema e na sua motivação. Neste caso, o problema reside na ausência de um *dashboard* em *Power BI* que centralize e automatize o *pipeline* de produção de métricas *DevOps*. Esta lacuna pode levar a que os utilizadores finais se sintam sobrecarregados com o volume de métricas dispersas por múltiplas ferramentas, resultando em decisões menos informadas ou até na falha de

DevOps and (Software) Operational Excellence Metrics

deteção de problemas no sistema. A falta de uma visão unificada e em tempo real dificultava a identificação de *bottlenecks*, o acompanhamento de KPIs e a tomada de decisão ágil. Assim, investigou-se uma forma de criar um *dashboard* eficaz, capaz de monitorizar a produtividade, o desempenho e a qualidade num ambiente *DevOps*, através de visualizações intuitivas e atrativas.

Tendo em conta o problema identificado e seguindo as etapas propostas, estabeleceu-se, na segunda fase, como objetivo principal o desenvolvimento de um dashboard em PowerBI. Este deverá seguir as boas práticas da comunidade e disponibilizar às equipas *DevOps* todas as métricas relevantes, permitindo uma monitorização eficaz e decisões mais fundamentadas acerca dos seus sistemas. Desta forma, pretende-se auxiliar os *stakeholders* no processo de tomada de decisão, fomentando a eficiência, a transparência e a melhoria contínua mediante a automatização e centralização das métricas.

Na terceira fase concebeu-se e desenvolveu-se o artefacto – um *dashboard* em *Power BI* com ligação à fonte de dados *Azure DevOps*, implementando visualizações interativas com DAX para cálculos complexos, além de garantir atualizações automáticas através do *Power BI Service*. O *dashboard*, desenvolvido para o grupo ANOVA, foi estruturado em quatro objetivos funcionais para otimizar a gestão de métricas *DevOps*:

1. Consolidação de dados – Agrega informações de múltiplas fontes (*Jira, GitHub, Azure DevOps*) num único repositório centralizado, eliminando a fragmentação de dados.
2. Automatização do pipeline ETL (Extração, Transformação e Carregamento) – Garante um fluxo de dados eficiente, fiável e sem intervenção manual.
3. Visualização intuitiva de métricas-chave – Apresenta os dados através de gráficos e *insights* de fácil interpretação, permitindo uma avaliação rápida do desempenho.
4. Suporte à tomada de decisão estratégica – Fornece dados em tempo real, atualizados automaticamente, que refletem com precisão o estado das operações.

Estes objetivos funcionais convergem para uma solução técnica robusta que não só agiliza o acesso à informação, mas também eleva significativamente a qualidade da análise de dados

DevOps and (Software) Operational Excellence Metrics

no contexto *DevOps*, proporcionando aos gestores e equipas uma ferramenta poderosa para monitorização contínua e melhoria de processos.

Na quarta fase – a demonstração do artefacto – foi feita a apresentação do artefacto ao supervisor do estágio para perceber se o mesmo era ou não a solução para resolver o problema apresentado. A apresentação do artefacto desenvolvido foi feita através de uma reunião remota via *Microsoft Teams*, onde se simularam cenários reais de utilização de modo a compreender se o *dashboard* desenvolvido era a melhor forma de apresentar as métricas *DevOps* seleccionadas. Para validar a sua utilidade prática, analisou-se especificamente a clareza na apresentação dos dados, a relevância das métricas escolhidas e a eficiência na identificação de problemas. Paralelamente, recolheu-se feedback detalhado para ajustar a hierarquia visual da informação, otimizar o layout e disposição dos elementos, refinar as funcionalidades interativas e melhorar a usabilidade geral do *dashboard*. Foi nesta fase que se percebeu se era possível avançar para a avaliação e conclusão do projeto, ou se ainda seria necessário realizar alguns ajustes e voltar a fase de desenvolvimento de objetivos e do artefacto.

De seguida, procedeu-se à avaliação dos resultados e da eficiência do artefacto desenvolvido. É importante comparar a funcionalidade do artefacto com os objetivos da solução propostos na segunda fase. No final da quinta fase (fase de avaliação), analisou-se a necessidade de retomar a terceira fase para melhorar a eficácia do artefacto ou de avançar para a comunicação dos resultados (Peffer et al., 2007), deixando melhorias adicionais para projetos futuros.

Por fim, na sexta fase, procedeu-se à comunicação dos resultados, documentando-se o processo de conceção e as respetivas implicações. Nesta etapa, foram ainda divulgados os resultados obtidos e apresentado o design com todas as funcionalidades finais. Um dos resultados desta fase de comunicação é o presente relatório de estágio de mestrado.

3 ENTIDADE ACOLHEDORA

Este capítulo apresenta as empresas Intelligent Sensing Anywhere, S.A. (ISA) e ANOVA, sendo que atualmente a ISA integra a estrutura da ANOVA, entidade responsável por acolher o estágio curricular. São abordadas a evolução das organizações, os seus princípios orientadores (missão, visão e valores), bem como as suas características distintivas.

3.1 Intelligent Sensing Anywhere, S.A.

Fundada em 1990 em Portugal, a ISA estabeleceu-se como empresa de referência no desenvolvimento de sistemas de telemetria para a indústria petrolífera e de gás natural. É uma empresa tecnológica pioneira em sistemas de telemetria e telegestão, com soluções inteligentes que integravam dados em plataformas de gestão para otimização de processos. Com operações em 32 países, especializou-se em monitorização remota para os setores energético e hidrocarbonetos, conquistando reconhecimento global.

A excelência técnica e a capacidade de inovação consolidaram a sua posição competitiva internacional. O modelo de atuação baseava-se na geração de valor em cadeia para clientes e respetivos *stakeholders* e adaptação contínua às exigências do setor. A ISA posicionava-se na liderança da revolução digital industrial, promovendo eficiência operacional através de integração de IoT na gestão energética, permitindo redes de sensores interconectados e aplicação de inteligência artificial para análise preditiva e automação avançada. Os seus pilares organizacionais assentavam em excelência técnica, inovação disruptiva e relacionamento excecional com clientes, sustentados por uma cultura de colaboração e melhoria contínua.

Em 2019, concretizou-se a sua aquisição pela ANOVA, conglomerado norte-americano com operações paralelas no mesmo segmento de mercado, configurando uma operação horizontal de consolidação empresarial. O processo de consolidação atingiu um novo patamar com a incorporação da ISA, marcando a transição estratégica para um modelo operacional integrado. Esta operação permitiu à ANOVA reforçar o seu capital humano, registando um aumento de 25 profissionais, o que elevou o total de colaboradores de 300 para 325.

3.2 ANOVA

A ANOVA é uma empresa líder a nível mundial em soluções de telemetria remota e análise de dados, destinada a setores como os gases industriais, produtos químicos, GPL, combustíveis e lubrificantes. Com presença em mais de 80 países e mais de um milhão de ativos monitorizados, a ANOVA permite às empresas otimizar as operações, reduzir custos e melhorar a sustentabilidade.

A história da ANOVA tem início em 1989, com a fundação da DataOnline em New Providence, New Jersey, por Robert Barnacle e Martin Adams, antigos colaboradores da BOC. Identificando uma oportunidade na monitorização de tanques criogénicos, a DataOnline destacou-se rapidamente, tendo a BOC como seu primeiro cliente na América do Norte.

Em 2016, a empresa de capital privado FFL Partners identificou uma oportunidade significativa no setor da tecnologia industrial e iniciou um processo estratégico de integração de várias empresas líderes, nomeadamente DataOnline, Wikon, WESROC, ISA, Silicon Controls e SDI. Este processo culminou, em 2019, com a criação da ANOVA. O nome “ANOVA” deriva do termo estatístico “análise de variância”, que simboliza o compromisso da empresa com a análise de dados e a inovação.

A ANOVA disponibiliza um conjunto abrangente de soluções para digitalizar cadeias de abastecimento, entre as quais se destacam:

- Unidades de Telemetria Remota (RTUs): Estas unidades monitorizam tanques estacionários, ativos móveis, compressores de CO₂, sistemas de mistura de gases, vaporizadores e outros equipamentos, oferecendo suporte a alertas personalizados e interfaces de entrada/saída adicionais.
- Suporte Avançado: A empresa presta apoio a outros ativos críticos da cadeia de abastecimento, tais como estações de abastecimento de hidrogénio e GNL, congeladores criogénicos, oleodutos, sistemas de tratamento térmico e sistemas PSA ou de membrana geridos por PLCs.

DevOps and (Software) Operational Excellence Metrics

- Software Anova Transcend®: Este software capta dados cruciais e amplia a visibilidade operacional, integrando diversos ativos em painéis de monitorização centralizados. Suporta a medição de nível de tanque, pressão, caudal, temperatura e outros parâmetros essenciais.
- ANOVA Asset Viewer Esta solução permite o acesso remoto à informação, facultando aos clientes a capacidade de tomar decisões proativas através de análises avançadas do desempenho dos ativos industriais – contribuindo assim para a redução dos custos de manutenção, do tempo de paragem e para a melhoria da eficiência logística.

A ANOVA adota um modelo flexível de Hardware como Serviço (HaaS), oferecendo hardware de monitorização, software, comunicações e suporte através de uma subscrição mensal. Este modelo elimina a necessidade de um investimento inicial elevado, permitindo substituições sem custos adicionais e proporcionando poupanças operacionais significativas.

As soluções da ANOVA ajudam os clientes a evitar ruturas de stock, a otimizar a mão de obra, a reforçar a segurança e a distribuir produtos com uma menor distância percorrida, contribuindo para a sustentabilidade ambiental global.

Esta abordagem consolida a ANOVA como um parceiro de confiança na gestão de ativos, não só nos últimos 35 anos como também para as próximas décadas. Se precisar de informações adicionais ou de uma adaptação para um público específico, estou disponível para ajudar.

4 PROJETO

Neste capítulo, apresenta-se o principal trabalho desenvolvido durante o estágio realizado na ANOVA, o qual dá nome ao próprio título do estágio: "*DevOps and (Software) Operational Excellence Metrics*". Este trabalho consistiu na conceção e construção de um artefacto – um *dashboard* em *Power BI* de métricas de excelência operacional em contextos *DevOps* – seguindo a metodologia DSR. Para tal, começou-se por identificar o problema: a dificuldade em consolidar e interpretar métricas *DevOps* dispersas por múltiplas plataformas, o que motivou o desenvolvimento de um *dashboard* único e automatizado. A solução proposta envolveu a transformação de dados em bruto (via *pipelines* ETL) e a utilização de indicadores-chave, recolhidos da revisão de literatura e enquadrados no contexto operacional do grupo ANOVA, de modo a apresentar a informação de forma clara e acionável. Por fim, são apresentados os *dashboards* desenvolvidos no *Power BI*, bem como as suas funcionalidades.

4.1 Identificação do Problema

Este projeto iniciou-se com a identificação das necessidades do grupo ANOVA, etapa fundamental para assegurar que a solução desenvolvida respondesse aos requisitos e desafios da organização. O primeiro passo consistiu em realizar uma reunião detalhada com o *Chief Software Architect* da ANOVA. Discutiram-se os objetivos estratégicos, os desafios operacionais existentes e as expetativas relativamente ao *dashboard* a desenvolver. Identificou-se uma necessidade premente de centralizar e integrar os dados, que se encontravam dispersos por múltiplos sistemas (*Azure DevOps*, *Jira*, *GitHub*, entre outros), com formatos heterogéneos e sem nomenclatura consistente, o que impossibilitava uma análise consolidada. Além disso, era crucial dispor de métricas *DevOps* dinâmicas e detalhadas, incluindo indicadores-chave como:

- Autonomia operacional (*Lead Time*, *Cycle Time*);
- Eficiência de *Deployments* (*Deployment Frequency*, *Change Failure Rate*);
- Qualidade de código (*Bug Rate*, *Test Coverage*);
- Resiliência operacional (*Mean Time to Recovery* – MTTR).

DevOps and (Software) Operational Excellence Metrics

Assim, a ANOVA necessitava de uma ferramenta que proporcionasse uma visão unificada e em tempo real do estado dos processos *DevOps*, aliada à capacidade de prever tendências e de suportar o planeamento com base em dados históricos e atuais. A usabilidade e o acesso simplificado à informação eram igualmente críticos, pelo que a solução teria de ser intuitiva e de fácil adoção por todos os níveis da organização – desde as equipas técnicas até à gestão. Era essencial que os dados fossem acessíveis e interpretáveis por utilizadores com diferentes perfis técnicos. Por fim, a segurança e privacidade dos dados constituíam uma prioridade absoluta, exigindo-se mecanismos robustos de proteção e conformidade com as regulamentações aplicáveis.

O projeto centrou-se, portanto, na resolução do problema enfrentado pelas equipas *DevOps* na monitorização e reporte de métricas, visando não só retratar o estado atual das operações de desenvolvimento, mas também extrair insights que permitissem comparações temporais e entre equipas. Era imperativo maximizar o potencial dos dados para suportar decisões estratégicas e antecipar cenários futuros. Deste modo, surgiu a necessidade de automatizar a extração e tratamento dos dados das diversas plataformas, consolidando-os num *dashboard* único em *Power BI*, que agregasse toda a informação relevante para todas as equipas. Verificou-se ainda que, para uma análise eficaz das métricas *DevOps*, era necessário considerar os processos mais relevantes para a ANOVA e selecionar os indicadores com maior impacto. Optou-se por adotar indicadores nas áreas da eficiência operacional, qualidade, velocidade e risco, com base na revisão de literatura, nas necessidades específicas da organização e nos indicadores mais relevantes utilizados em reuniões de acompanhamento.

4.2 Definição dos Objetivos

Após a deteção dos problemas, procedeu-se à definição dos objetivos, tendo em conta as necessidades identificadas para a monitorização e melhoria dos processos *DevOps*. Através das reuniões realizadas com o supervisor, identificaram-se as melhorias aplicáveis, descritas em seguida.

DevOps and (Software) Operational Excellence Metrics

O objetivo geral deste projeto consistiu na criação de uma solução que permitisse a extração, transformação e visualização de dados sobre os processos *DevOps*. Especificamente, desenvolveram-se:

- Fluxos ETL em *Power Query* para extrair e tratar os dados das diversas plataformas (*Azure DevOps, Jira, GitHub*);
- Cálculos avançados com DAX para derivar métricas complexas;
- *Dashboards* interativos em *Power BI* para apresentação dos resultados.

A solução visou cumprir os objetivos estratégicos definidos pela gestão da ANOVA:

- Melhorar a monitorização contínua dos processos *DevOps*;
- Fornecer *feedback* em tempo real às equipas e gestores;
- Dinamizar reuniões com base em dados atualizados;
- Simplificar a interpretação de métricas técnicas para públicos não especializados.

Para garantir a concretização dos objetivos, implementaram-se as seguintes ações:

- Transformação de Dados
 - Limpeza e normalização dos dados em *Power Query*;
 - Criação de um modelo dimensional otimizado para análise.
- Automatização do Processo
 - Configuração de atualizações automáticas via *Power BI Service*;
 - Eliminação de processos manuais na preparação de relatórios.
- Funcionalidades Avançadas
 - Filtros para análise detalhada por projeto, equipa ou período;

Salienta-se que o *dashboard* desenvolvido deu prioridade à clareza e usabilidade, garantindo que utilizadores sem conhecimentos técnicos em *DevOps* pudessem interpretar as métricas de forma intuitiva.

4.3 Desenvolvimento do Artefacto

O desenvolvimento do artefacto – um *dashboard* em *Power BI* – seguiu uma abordagem estruturada e com os requisitos identificados nas fases anteriores. O processo dividiu-se em etapas fundamentais, desde a extração e transformação de dados até à implementação de visualizações interativas, assegurando uma solução robusta e adaptada às necessidades do grupo ANOVA.

4.3.1 Conceção

Com a definição dos objetivos, face aos problemas e necessidades identificados no grupo ANOVA, procedeu-se à tradução dos requisitos em especificações técnicas claras para a solução de BI. Este processo incluiu as seguintes etapas:

1. Definição das fontes de dados:
 - Identificação e seleção das plataformas a integrar (*Azure DevOps*, *Jira* e *GitHub*), assegurando a cobertura de todas as métricas *DevOps* relevantes;
 - Mapeamento dos campos necessários para o cálculo de indicadores-chave (*Lead Time*, *Cycle Time*, *Deployment Frequency*, entre outros).
2. Desenvolvimento do modelo de dados
 - Criação de um modelo multidimensional no *Power BI*, otimizado para análise de desempenho;
 - Estabelecimento de relações entre tabelas de factos (ex.: *deployments*, *work items*) e dimensões (ex.: tempo, equipas, projetos);
 - Normalização dos dados para garantir consistência e fiabilidade.
3. Desenho de visualizações intuitivas
 - Conceção de *mockups* iniciais, validados com os *stakeholders*, para assegurar clareza e usabilidade;
 - Definição de KPI prioritários e respetivas representações visuais (gráficos de tendência, tabelas dinâmicas, indicadores de estado);
 - Implementação de filtros interativos para análise segmentada (por projeto, equipa ou período).

4. Automatização e segurança

- Configuração de *pipelines* ETL no *Power Query* para extração e transformação automática de dados;
- Implementação de mecanismos de segurança, incluindo controlo de acessos e conformidade com as políticas de proteção de dados.

Esta fase garantiu que o *dashboard* não só respondesse às necessidades operacionais, como também oferecesse uma experiência analítica robusta, alinhada com as melhores práticas de *Data Visualization* e *DevOps Analytics*.

4.3.1.1 Métricas *DevOps*

Antes de definir as métricas mais relevantes segundo os padrões da indústria, na reunião inicial com o *Chief Software Architect* da ANOVA, foram identificados os indicadores de desempenho *DevOps* considerados críticos para a monitorização operacional. Debateu-se quais eram as métricas mais significativas para acompanhar a eficiência dos processos, bem como que dados poderiam ser extraídos das plataformas existentes (*Azure DevOps*, *Jira*, *GitHub*) para suportar essa análise.

Para validar as escolhas iniciais, efetuou-se uma revisão bibliográfica, que serviu de base científica para a seleção final das métricas. Nesta fase, priorizaram-se quatro métricas fundamentais:

1. Frequência de Implementações (*Deployment Frequency*):

Esta métrica mede quantas vezes a equipa coloca código em produção num determinado período de tempo (por exemplo, diariamente, semanalmente ou por sprint). Refere-se à regularidade com que funcionalidades, melhorias ou correções são efetivamente disponibilizadas aos utilizadores. Reflete a cadência de entrega e a agilidade da equipa no fornecimento de valor ao utilizador. Permite também avaliar se a equipa está a conseguir entregar pequenas alterações de forma incremental, o que tende a reduzir riscos e facilitar o feedback rápido. Uma frequência de implementação elevada, acompanhada de qualidade, é indicadora de processos de integração e entrega contínua bem implementados. Em suma, ajuda a perceber se a

DevOps and (Software) Operational Excellence Metrics

organização está preparada para responder rapidamente às mudanças do mercado, garantindo entregas regulares, previsíveis e seguras. Em *DevOps*, este indicador é uma referência da maturidade operacional e técnica (Fu et al., 2025).

2. Taxa de Defeitos em Produção (*Defect Escape Rate*):

Esta métrica indica a proporção de erros ou falhas detetadas após a colocação em produção, ou seja, defeitos que não foram identificados durante o desenvolvimento ou os testes internos. Avalia diretamente a eficácia do processo de controlo de qualidade. Uma taxa elevada de defeitos aponta para potenciais falhas na cobertura de testes, na validação funcional ou na revisão de código. Enquanto, uma taxa reduzida de defeitos sugere que o sistema de testes é fiável e que as alterações estão a ser cuidadosamente validadas antes de serem lançadas. Resumindo, defeitos em produção impactam negativamente a experiência do utilizador, a imagem da organização e podem originar custos significativos associados à correção urgente e à perda de confiança dos clientes. Esta métrica é fundamental para garantir a qualidade e estabilidade das entregas (Amaro et al., 2024).

3. Tempo de Ciclo (*Cycle Time*):

O tempo de ciclo corresponde ao intervalo entre o início do desenvolvimento de uma tarefa (por exemplo, quando um programador começa a trabalhar numa funcionalidade) até o momento em que essa tarefa está pronta para ser colocada em produção. Permite medir a eficiência do processo de desenvolvimento propriamente dito. Por outro lado, ajuda a identificar gargalos internos, como excesso de trabalho em curso, atrasos em revisões ou processos manuais demorados. Um tempo de ciclo reduzido sugere que a equipa é ágil na execução técnica, sem bloqueios significativos nas fases de codificação, revisão de código ou testes automatizados. Resumindo, é uma métrica operacional chave para equipas técnicas, que permite melhorar continuamente os fluxos de trabalho e reduzir tempos de entrega sem comprometer a qualidade (Bhardwaj et al., 2023).

4. Tempo de Espera (*Lead Time*):

O *lead time* mede o tempo total que decorre desde o momento em que uma funcionalidade ou tarefa é solicitada (ou identificada como necessidade) até ao

DevOps and (Software) Operational Excellence Metrics

momento em que é efetivamente disponibilizada em produção. Esta métrica abrange todas as etapas do processo, incluindo:

- Recolha e análise de requisitos
- Priorização e planeamento
- Desenvolvimento
- Testes
- Entrega em produção

Diferencia-se do *cycle time* por incluir todo o percurso desde a conceção até à disponibilização ao utilizador final. O *lead time* avalia a eficiência global do fluxo de trabalho, desde a ideia inicial até à entrega de valor. Ajuda também a identificar estrangulamentos fora da execução técnica, como tempos de espera em fases de planeamento, aprovação ou alocação de recursos. Um *lead time* reduzido significa que a organização é capaz de responder rapidamente às necessidades do mercado ou dos clientes. Concluindo, é uma das métricas mais abrangentes e estratégicas em *DevOps*, pois reflete a capacidade da organização de entregar valor de forma contínua e eficiente. É essencial para o alinhamento entre equipas técnicas e de negócio (Li et al., 2022).

Estas quatro métricas, quando analisadas em conjunto, oferecem uma visão completa e equilibrada da performance e maturidade *DevOps* da equipa ou organização, permitindo:

- Monitorizar e melhorar continuamente a eficiência dos processos de desenvolvimento;
- Identificar estrangulamentos e áreas críticas ao longo do fluxo de trabalho;
- Assegurar um equilíbrio saudável entre velocidade, qualidade e fiabilidade;
- Reforçar a capacidade de adaptação às mudanças e de entrega contínua de valor aos utilizadores finais.

Estas métricas não devem ser vistas de forma isolada, mas como componentes de um sistema interligado de medição de desempenho *DevOps*, essencial para suportar uma cultura de melhoria contínua, colaboração entre equipas e entrega ágil e sustentável.

Paralelamente, foi essencial compreender o público-alvo do *dashboard* (equipas técnicas, gestores de projeto e *stakeholders* não técnicos) e o seu nível de familiaridade com conceitos *DevOps*. O objetivo era garantir que as visualizações fossem intuitivas, permitindo uma interpretação rápida e precisa por parte de todos os utilizadores. A seleção dos tipos de gráficos (colunas agrupadas, cartões de KPI e gráficos circulares) teve como objetivo facilitar comparações temporais, entre equipas e entre métricas, assegurando clareza na comunicação dos dados. Este cuidado estendeu-se também à escolha das métricas em si, que foram integradas no *dashboard* em total alinhamento com as categorias de desempenho do DORA (*Elite, High, Medium, Low*) e com os critérios da ANOVA. Desta forma, o *dashboard* não só permitia uma leitura intuitiva dos dados, mas também uma avaliação imediata do desempenho face a *benchmarks* setoriais e objetivos internos. Esta abordagem garantiu que o *dashboard* não apenas refletisse o estado atual dos processos, mas também identificasse oportunidades de melhoria, alinhadas com as melhores práticas do setor e com as necessidades específicas da ANOVA.

4.3.1.2 Modelo Multidimensional

Ainda na fase de conceção, definiu-se o modelo multidimensional a implementar no *Power BI*, incluindo a estrutura das tabelas, os respetivos relacionamentos e as direções associadas. Os esquemas mais comuns no *Power BI* são o esquema em estrela e o esquema em floco de neve. Optou-se pelo esquema em estrela, por ser o mais consensual entre os utilizadores e o mais recomendado (Maslyuk, 2022).

Neste modelo, as tabelas classificam-se em dois tipos:

- Tabelas de facto: Contêm as métricas a agregar e incluem chaves estrangeiras para relacionamento com as dimensões.
- Tabelas de dimensão: Contêm atributos descritivos que permitem analisar as tabelas de facto. Cada dimensão possui uma chave primária (identificador único) e colunas descritivas.

No modelo adotado, as tabelas de facto seguem o padrão estabelecido e são relacionadas com as tabelas de dimensão, normalmente através de relações muitos-para-um. Assim, na

fase inicial de desenvolvimento do projeto, foi concebido o modelo multidimensional (Figura 4.1) que serviu de base para a implementação do projeto e criação do *dashboard* em *Power BI*.

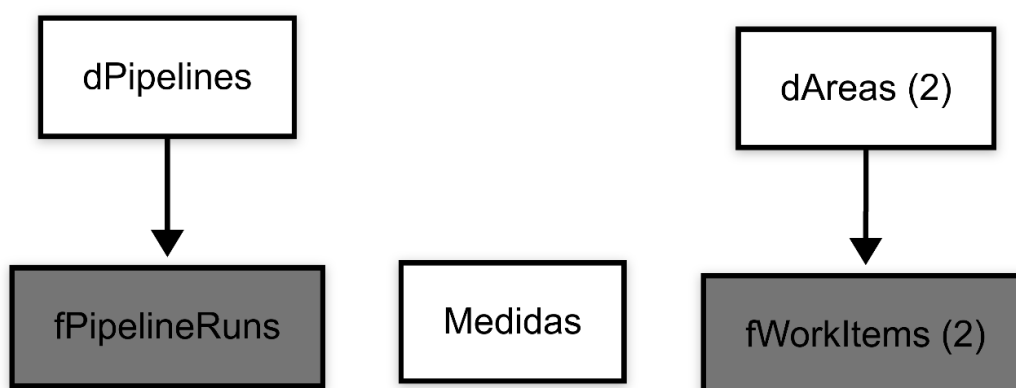


Figura 4.1. Desenho do modelo multidimensional no esquema em estrela

Dado que o modelo será implementado no *Power BI* detalha-se na fase de implementação as tabelas e respetivos os atributos, bem como as medidas criadas.

4.3.1.3 Mockups do Dashboard

Ainda durante a fase de conceção, foi importante definir a forma como os dados seriam apresentados e como os utilizadores interagiriam com o *dashboard*. Para tal, a criação de *mockups* revelou-se uma etapa fundamental, permitindo definir e visualizar a disposição dos elementos visuais antes da implementação final.

Idealizou-se um conjunto de quatro *dashboards* independentes, cada um dedicado a uma métrica fundamental do ciclo *DevOps*, permitindo uma análise granular e especializada de cada aspeto do processo de desenvolvimento e operações. Esta abordagem modular foi adotada para garantir uma focagem absoluta em cada indicador-chave, evitando a dispersão de informação e facilitando a identificação rápida de problemas específicos.

Na Figura 4.2 apresenta-se a arquitetura global da solução, onde se verifica a divisão lógica entre os quatro módulos analíticos.

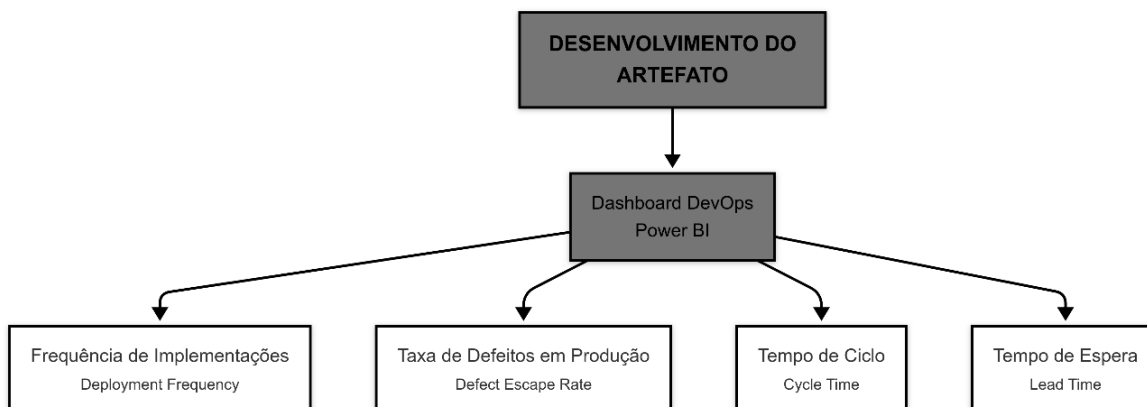


Figura 4.2. Artefacto a desenvolver em Power BI

O primeiro *dashboard*, dedicado à Frequência de Implementações (*Deployment Frequency*), foi concebido para monitorizar a cadência de *releases*, elemento crítico para avaliar a maturidade dos processos de integração e entrega contínua. Neste painel, os indicadores principais incluem: o volume total de implementações, a média diária de *deploys*, a distribuição temporal (diária e mensal) e a identificação de padrões sazonais. A visualização combina gráficos de colunas agrupadas para mostrar a evolução temporal, complementados por indicadores numéricos que destacam os valores mais recentes nos últimos 28 dias.

O segundo *dashboard*, focado na Taxa de Defeitos em Produção (*Defect Escape Rate*), foi estruturado para avaliar a eficácia dos processos de controlo de qualidade. Aqui, a análise centra-se na percentagem de erros que atingem ambiente produtivo, um indicador crucial da robustez dos testes automatizados e da revisão de código. Este painel incorpora: uma série temporal da taxa mensal de defeitos em produção, um *breakdown* da origem dos problemas (produção vs. ambientes não produtivos), e uma análise comparativa entre equipas ou componentes do sistema. A representação gráfica utiliza um gráfico de colunas agrupadas para a evolução histórica, complementado por um gráfico circular para mostrar a proporção atual de *bugs* em diferentes ambientes, detalhando a natureza e severidade de cada incidente.

O terceiro módulo, dedicado ao Tempo de Ciclo (*Cycle Time*), foi desenhado para medir a eficiência do fluxo de trabalho de desenvolvimento. Este *dashboard* apresenta uma análise multidimensional que inclui: a média histórica do tempo necessário para concluir tarefas desde o *commit* até à implementação, e a identificação de *outliers* que possam indicar

bloqueios processuais. A visualização principal utiliza gráficos de colunas agrupadas e cartões, permitindo identificar tanto tendências de longo prazo como variações pontuais.

O quarto e último *dashboard*, centrado no Tempo de Espera (*Lead Time*), foi concebido para analisar o tempo total desde a criação do *ticket* até à resolução efetiva em produção, incluindo períodos de espera e processamento. Este painel diferencia-se por focar-se nos tempos totais do processo, com indicadores que incluem: o *lead time* médio por tipo de tarefa, a comparação entre equipas ou projetos, e a análise dos componentes do tempo total (tempo ativo vs. tempo em espera). A representação visual utiliza gráficos de colunas agrupadas e cartões para decompor e identificar padrões temporais e gargalos processuais.

Na Figura 4.3 mostra-se o *mockup* inicial do *dashboard* de Frequência de Implementações, onde se pode observar a disposição espacial dos componentes: os gráficos principais ocupam a zona central, ladeados pelos filtros e pelos indicadores-chave à direita. A paleta cromática foi escolhida para garantir uma boa distinção entre diferentes tipos de informação, com cores quentes (vermelhos/laranjas) para alertas e cores frias (azuis/verdes) para indicadores positivos.

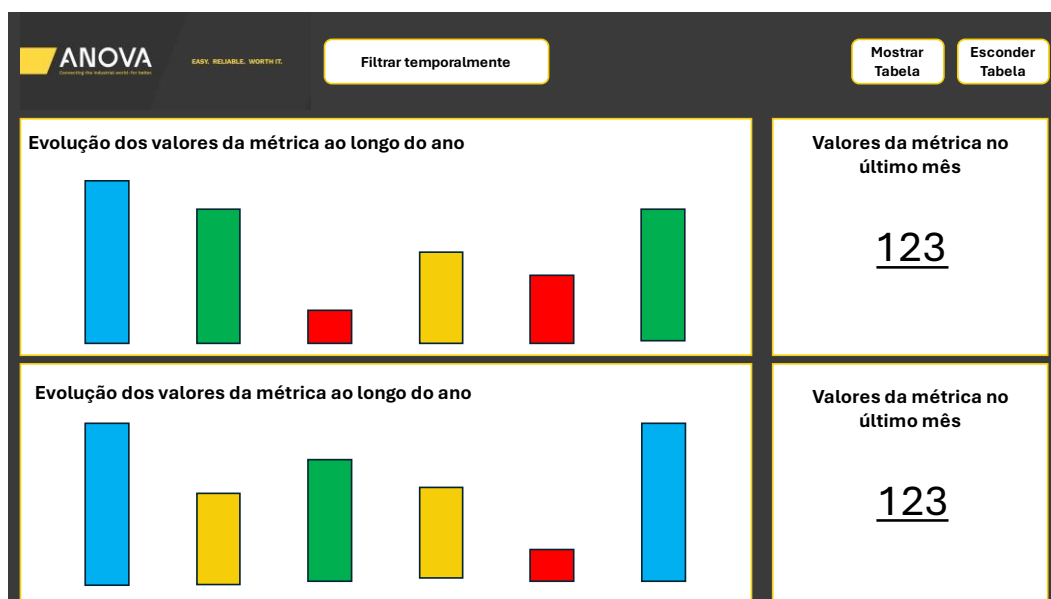


Figura 4.3. Mockup para o dashboard “Frequência de Implementações (Deployment Frequency)”

Cada *dashboard* incorpora um sistema completo de filtragem e segmentação, permitindo aos utilizadores: selecionar períodos temporais específicos, filtrar por equipa ou projeto, e focar-

DevOps and (Software) Operational Excellence Metrics

se em componentes específicos do sistema. Estes controlos de filtragem estão posicionados de forma consistente em todos os *dashboards*, criando uma experiência de utilização uniforme e intuitiva. A segurança dos dados foi garantida através da implementação de *Row-Level Security* (RLS) no *Power BI*, assegurando que cada utilizador só tem acesso aos dados relevantes para o seu âmbito de atuação.

4.3.2 Implementação

Nesta secção, descreve-se o desenvolvimento do *dashboard DevOps* em *Power BI*, com enfoque na extração, transformação e carregamento (ETL) de dados provenientes de fontes como o *Azure DevOps*, *Jira* e *GitHub*. A implementação das métricas previamente discutidas foi realizada recorrendo à linguagem DAX e à ferramenta *Power BI*, permitindo a criação de visualizações dinâmicas e automatizadas que refletem o desempenho dos processos *DevOps*. Como se verá adiante, a utilização do *Power Query* e de medidas em DAX possibilitou automatizar o tratamento dos dados, garantindo a precisão e a atualização contínua da informação apresentada no *dashboard*.

4.3.2.1 Extract, Transform and Load

O processo de ETL permite integrar dados provenientes de múltiplas fontes, como o *Azure DevOps*, *Jira* e *GitHub*.

Inicialmente, os dados encontravam-se dispersos por diferentes plataformas, cada uma com estrutura e formato próprios, o que representava um desafio na consolidação das métricas *DevOps* num modelo de dados unificado. O objetivo passava por agregar toda a informação num formato padronizado, facilitando a sua importação para o *Power BI* e garantindo uma análise eficiente. Para o efeito, foi necessário normalizar os dados, assegurando não só a consistência dos campos, mas também a correlação entre as diversas fontes. Optou-se pela utilização do *Power Query* para efetuar a transformação e tratamento dos dados, dado tratar-se de uma ferramenta integrada no *Power BI* que permite automatizar todo o processo ETL. Foram desenvolvidos fluxos de dados para extrair, limpar e estruturar a informação, incluindo operações de conversão de formatos, padronização de nomenclaturas e estabelecimento de relações entre tabelas. A extração de dados foi realizada através de

DevOps and (Software) Operational Excellence Metrics

uma API *OData* fornecida pelo *Azure DevOps* da ANOVA. Este protocolo permite consultar e manipular dados de forma estruturada, seguindo o padrão REST. Após estabelecer a conexão, foi criado um modelo semântico chamado “*DevopsMetrics*”, que contém as tabelas necessárias para análise.

Tabela 4.1. Tabelas importadas do modelo semântico “*DevopsMetrics*”

Tabela	Descrição
Areas (2)	Contém informações sobre áreas de trabalho ou equipas.
PipelineRuns	Registos de execuções de <i>pipelines</i> (sucessos, falhas, durações).
Pipelines	Definições dos <i>pipelines</i> (nomes, configurações, <i>triggers</i>).
WorkItems (2)	Itens de trabalho (tarefas, <i>bugs</i> , <i>user stories</i>).

Transformações Iniciais:

- Seleção de colunas relevantes: Eliminação de campos desnecessários para otimizar o modelo.
- Tipos de dados: Conversão de textos para categorias, datas para formato *datetime*, etc.
- Normalização de nomes: Padronização de nomes de colunas (ex: “*CreatedDate*” em vez de “*Date_Creation*”).

O processo de extração e importação seguiu uma abordagem estruturada:

1. Conexão segura via *OData* + *OAuth2* para garantir acesso autorizado;
2. Definição clara do modelo semântico, com tabelas otimizadas para análise *DevOps*;
3. Transformações para garantir qualidade e usabilidade dos dados;
4. Configuração de atualização conforme necessidades operacionais.

Este processo estruturado de extração e transformação de dados permitiu criar uma base analítica robusta que:

1. Centraliza todas as métricas *DevOps* críticas numa plataforma única de referência.
2. Automatiza o pipeline de dados, eliminando processos manuais propensos a erros.
3. Permite análises temporais através de dados históricos consistentes.
4. Suporta decisões *data-driven* com indicadores atualizados em tempo real.

A solução implementada não só cumpre os requisitos atuais de monitorização, como foi concebida para:

- Escalar com a adição de novas fontes de dados;
- Evoluir com a incorporação de métricas adicionais;
- Adaptar-se a mudanças nos processos *DevOps* da ANOVA.

Através deste *dashboard*, a ANOVA dispõe agora de uma ferramenta estratégica que transforma dados brutos do *Azure DevOps* em *insights* acionáveis, contribuindo diretamente para a melhoria contínua dos seus processos de desenvolvimento e entrega de *software*.

A solução final proporciona à ANOVA uma base analítica confiável para decisões *data-driven* no domínio *DevOps*, com capacidade de evolução para incorporar novas fontes e métricas. Este fluxo assegurou que os dados fossem fiáveis, atualizáveis e prontos para visualização no *Power BI*.

4.3.2.2 Modelo Multidimensional

Como definido na fase de conceção (subcapítulo 4.3.1.2), o modelo multidimensional implementado segue o esquema em estrela, otimizado para suportar o *dashboard* em *Power BI* com dados *DevOps*. Esta estrutura organiza-se em tabelas de facto e de dimensão, garantindo relações claras e eficiência analítica (Figura 4.4).

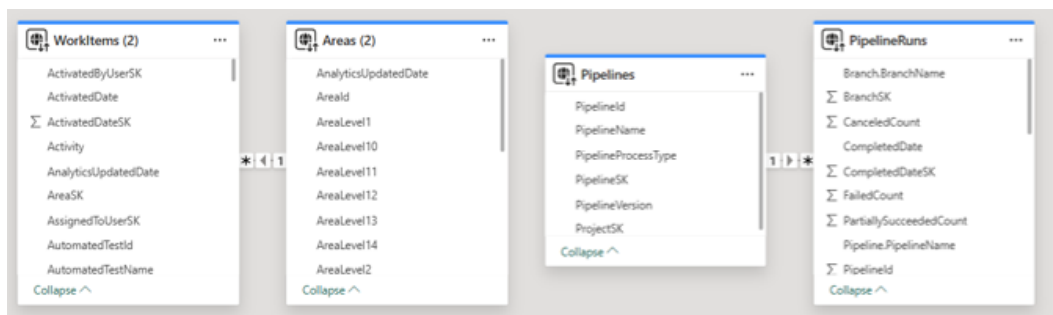


Figura 4.4. Modelo Multidimensional

A tabela de facto *WorkItems* regista informações detalhadas sobre itens de trabalho num sistema como o *Azure DevOps*, funcionando como base para análises e relatórios. Inclui campos como datas-chave (criação, fecho, conclusão), identificadores de utilizadores (quem criou, atribuiu ou alterou o item), e relacionamentos com áreas, iterações e revisões de código. Os campos com sufixo SK (*surrogate keys*) ligam a tabelas de dimensão, permitindo análises multidimensionais. Além disso, contém campos personalizados (como ambiente e revisor de design) e dados específicos de revisão de código (como aprovador e *GitHub User ID*). Esta estrutura permite rastrear o ciclo de vida dos itens, medir eficiência e gerar métricas sobre o progresso do trabalho.

A tabela de facto *PipelineRuns* regista execuções de pipelines de CI/CD (Integração Contínua/Entrega Contínua), contendo campos como *BranchName* (nome do *branch* Git associado), *RunNumber* (identificador da execução), *RunOutcome* (resultado: sucesso, falha, etc.) e *RunReason* (motivo da execução: *manual*, *trigger*, *scheduled*). Inclui ainda campos temporais (*Year*, *Quarter*, *Month*, *Day*) para análise temporal. Esta estrutura permite monitorizar eficiência, taxas de sucesso e padrões de execução ao longo do tempo, suportando análises como tendências de falhas ou impacto de *branches* específicos.

A tabela de dimensão *Areas* organiza a hierarquia de áreas (ex.: departamentos, equipas, módulos) num sistema de gestão de projetos, como o *Azure DevOps*. Inclui campos como *AreaSK* (chave substituta), *AreaId* (identificador original), *AreaName* (nome da área) e *AreaPath* (caminho hierárquico completo). A estrutura hierárquica é detalhada em níveis (*AreaLevel1* a *AreaLevel14*), permitindo flexibilidade para diferentes profundidades de categorização (ex.: "Empresa > Departamento > Equipa > Projeto"). Campos temporais

DevOps and (Software) Operational Excellence Metrics

(*Year, Quarter, Month, Day*) registam quando a informação foi válida, suportando análises históricas.

A tabela de dimensão *Pipeline* cataloga todos os pipelines disponíveis no sistema, com informações como nome, versão e tipo de processo. Através da chave *PipelineSK*, permite contextualizar as execuções registadas na tabela de facto *PipelineRuns* dentro do projeto correto, suportando análises detalhadas por *pipeline* específico ou grupo de *pipelines*.

A tabela de facto *WorkItems* regista informações detalhadas sobre itens de trabalho num sistema como o *Azure DevOps*, funcionando como base para análises e relatórios. Inclui campos como:

- Datas-chave: *ActivateiDate* (ativação de itens), *AnalyticsUpdateiDate* (atualizações analíticas)
- Identificadores: *AutomatedTestId* e *AutomatedTestName* (para rastreio de testes automatizados)
- Relacionamentos: *AreaSK* (ligação à dimensão *Areas*)

Os campos com sufixo SK (como *AreaSK*) representam chaves substitutas que ligam às tabelas de dimensão correspondentes, permitindo análises multidimensionais. Esta estrutura suporta o rastreio do ciclo de vida dos itens, medição de eficiência e geração de métricas de progresso.

A tabela de facto *PipelineRuns* regista execuções de *pipelines* de CI/CD, contendo:

- Contexto de execução: *Branch.BranchName* (*branch* Git associado), *Pipeline.PipelineName* (referência ao *pipeline*)
- Resultados: *CanceledCount*, *FailedCount*, *PartiallySuccesdedCount* (métricas de sucesso/fracasso)
- Datas: *CompletedDate* (data de conclusão)
- Relacionamentos: *PipelineSK* (ligação à dimensão *Pipelines*)

Campos como *BranchSK* e *PipelineSK* permitem cruzar dados com outras dimensões, enquanto a estrutura hierárquica (ex.: contagem de falhas por *branch*) facilita análises de eficiência e padrões temporais.

DevOps and (Software) Operational Excellence Metrics

A tabela de dimensão *Areas* organiza hierarquias de áreas (ex.: departamentos, equipas) com:

- Estrutura flexível: *AreaLevel1* a *AreaLevel14* (até 14 níveis de profundidade)
- Identificadores: *AreaId* (ID original), *AnalyticsUpdateiDate* (sincronização com atualizações)
- Relacionamentos: *ProjectSK* (vinculação a projetos)

Esta dimensão suporta análises históricas (via campos temporais) e integra-se com *WorkItems* através de chaves substitutas (ex.: *AreaSK*).

A tabela de dimensão *Pipelines* cataloga *pipelines* com:

- Metadados: *PipelineName*, *PipelineProcessType*, *PipelineVersion*
- Relacionamentos: *ProjectSK* (ligação a projetos), *PipelineSK* (chave substituta)

Esta dimensão contextualiza as execuções da tabela *PipelineRuns*, permitindo análises granulares por pipeline ou grupo de *pipelines* e integra-se com *PipelineRuns* através de chaves substitutas (ex: *PipelineSK*).

Relações Principais:

- *Areas* → *WorkItems*: Via campo *AreaSK*, permitindo filtrar itens por hierarquia organizacional.
- *Pipelines* → *PipelineRuns*: Através de *PipelineSK*, associando execuções à definição do pipeline.
- Consistência dimensional: Todas as tabelas usam *ProjectSK* para análises cruzadas por projeto.

Este modelo multidimensional, representado na Figura 4.4, oferece uma base de dados robusta para o *dashboard Power BI*, alinhando perfeita integridade técnica com as necessidades analíticas *DevOps*. Através da combinação estratégica de tabelas de facto ricas em métricas com dimensões descritivas bem estruturadas, possibilita uma análise compreensiva do desempenho de engenharia, desde a granularidade de itens individuais até visões agregadas por equipas, projetos e períodos temporais.

4.3.2.3 Criação de medidas usando DAX

Com os dados tratados, recorreu-se à linguagem DAX para criar medidas. Trata-se da linguagem de programação do *Power BI*, que permite realizar cálculos variáveis e aplicar condições (Russo & Ferrari, 2019). A utilização desta linguagem é fundamental, pois otimiza o desempenho e reduz o tamanho do modelo de dados. Embora seja possível agregar colunas diretamente em elementos visuais sem recorrer a medidas explícitas, a utilização de DAX é geralmente mais vantajosa, dado que melhora a performance do *Power BI*.

Importa esclarecer a distinção entre medidas implícitas (geradas automaticamente pelo sistema, podendo originar valores inesperados) e medidas explícitas (criadas manualmente pelo utilizador através de fórmulas). Neste projeto, optou-se exclusivamente por medidas explícitas. De seguida, enumeram-se as funções DAX utilizadas na criação das medidas, cuja elaboração seguiu as orientações recomendadas (Russo & Ferrari, 2019).

A medida DAX apresentada na Figura 4.5 tem como objetivo contar o número de implementações (*deploys*) bem-sucedidas com base nos registos armazenados numa tabela denominada *PipelineRuns*. O nome atribuído à medida é "*Number of Deploys*", o qual será utilizado para referenciar este cálculo em relatórios ou visualizações.

A lógica da medida está estruturada em duas partes principais. Primeiro, é declarada uma variável chamada *passed*, que armazena um cálculo intermédio. Esta variável utiliza a função *CALCULATE*, que modifica o contexto de filtro para contar apenas as linhas da tabela *PipelineRuns* que correspondem a um critério específico. Dentro desta função, *COUNTROWS(PipelineRuns)* conta o número total de registos na tabela, enquanto *FILTER* restringe essa contagem apenas aos casos em que a coluna *[RunOutcome]* contém o valor "*Succeed*". A verificação é feita através da função *CONTAINS*, que procura a palavra exata dentro do texto da coluna.

A segunda parte da medida começa com a instrução *RETURN*, que define o resultado final. Aqui, é aplicada uma condição utilizando *IF* e *ISBLANK* para garantir que, caso a variável *passed* esteja vazia (ou seja, não existam registos com "*Succeed*"), o retorno seja 0 em vez de um valor nulo. Caso contrário, a medida devolve o valor calculado em *passed*.

DevOps and (Software) Operational Excellence Metrics

Em resumo, esta medida foi concebida para monitorizar o sucesso de implementações num pipeline, contabilizando apenas as execuções com o resultado "Succeed" e assegurando que o resultado seja sempre um número válido (mesmo que zero) para evitar erros em visualizações ou *dashboards*.

```
1 Number of Deploys =  
2 |   VAR passed = CALCULATE(  
3 |   |   |   |   |   COUNTROWS(PipelineRuns),  
4 |   |   |   |   |   FILTER(PipelineRuns, CONTAINS(PipelineRuns,[RunOutcome],"Succeed"))  
5 |   )  
6 RETURN IF( ISBLANK(passed), 0, passed)
```

Figura 4.5. Exemplo de um extrato do código em DAX para contagem de implementações bem-sucedidas

A medida apresentada na Figura 4.6, denominada "Average Number of Deploys per Day", tem como objetivo calcular a média diária de implementações (*deploys*) com base nos registos da tabela *PipelineRuns*.

A lógica da medida é simples e direta, utilizando a função *COUNTA* para contar o número de entradas não vazias na coluna *[RunOutcome]* da tabela *PipelineRuns*. Esta função difere de *COUNT* por incluir todos os tipos de dados (não apenas valores numéricos), o que é útil quando se trabalha com colunas de texto ou resultados categóricos, como é o caso.

O resultado deste cálculo é depois dividido por 28, assumindo-se que este valor representa o número total de dias no período em análise. Por exemplo, se os dados abrangerem quatro semanas (28 dias), a medida calcula quantos *deploys* ocorreram, em média, por dia. Esta abordagem é comum em métricas de desempenho de pipelines de CI/CD, onde se pretende avaliar a frequência média de implementações ao longo de um ciclo temporal definido.

Em resumo, esta medida oferece uma visão agregada da cadência média de implementações, sendo útil para identificar padrões de produtividade ou necessidades de otimização em processos de desenvolvimento.

```
1 Average Number of Deploys per Day =  
2 COUNTA('PipelineRuns'[RunOutcome]) / 28
```

Figura 4.6. Exemplo de fórmula em DAX para média de implementações por dia, utilizando a função *COUNTA* e divisão por período fixo (28 dias)

A medida "*Bugs Production*" apresentada na Figura 4.7 tem como objetivo contabilizar o número de *bugs* reportados em ambiente de produção, com base nos registos da tabela '*WorkItems (2)*'. Esta métrica é particularmente relevante para equipas de desenvolvimento e operações, permitindo monitorizar a qualidade do *software* em ambiente real e identificar necessidades de correção ou melhoria.

A estrutura da medida segue um padrão comum em DAX, começando pela declaração de uma variável chamada *bugsprod*, que armazena o resultado intermédio do cálculo. Esta variável utiliza a função *CALCULATE*, que modifica o contexto de filtro para contar apenas as linhas da tabela '*WorkItems (2)*' que satisfazem uma condição específica. A função *COUNTROWS* é empregue para contar o número total de registos na tabela, enquanto a condição *CONTAINSSTRING* verifica se a coluna '*Custom_Environment*' contém a palavra "*Production*". Esta abordagem assegura que apenas os *bugs* associados ao ambiente de produção sejam incluídos na contagem.

A segunda parte da medida, iniciada pela instrução *RETURN*, garante que o resultado final seja sempre um valor numérico válido, mesmo que não existam *bugs* registados em produção. A função *IF* combinada com *ISBLANK* verifica se a variável *bugsprod* está vazia e, em caso afirmativo, retorna 0, evitando assim a exibição de valores nulos em relatórios ou *dashboards*. Caso contrário, o valor calculado em *bugsprod* é devolvido.

Esta medida destaca-se pela sua robustez e clareza, sendo especialmente útil para equipas que necessitam de acompanhar de perto a estabilidade de aplicações em produção. Em resumo, a medida "*Bugs Production*" oferece uma forma eficaz de quantificar problemas em ambiente de produção, facilitando a identificação de tendências e a priorização de correções, contribuindo assim para a melhoria contínua da qualidade do *software*.

```
1 Bugs Production =
2     VAR bugsprod = CALCULATE(
3         |         |         |         |         |         |         |         |         |         |
4         |         |         |         |         |         |         |         |         |         |
5         |         |         |         |         |         |         |         |         |         |
6         |         |         |         |         |         |         |         |         |         |
7         |         |         |         |         |         |         |         |         |         |
8         |         |         |         |         |         |         |         |         |         |
9         |         |         |         |         |         |         |         |         |         |
10        |         |         |         |         |         |         |         |         |         |
11        |         |         |         |         |         |         |         |         |         |
12        |         |         |         |         |         |         |         |         |         |
13        |         |         |         |         |         |         |         |         |         |
14        |         |         |         |         |         |         |         |         |         |
15        |         |         |         |         |         |         |         |         |         |
16        |         |         |         |         |         |         |         |         |         |
17        |         |         |         |         |         |         |         |         |         |
18        |         |         |         |         |         |         |         |         |         |
19        |         |         |         |         |         |         |         |         |         |
20        |         |         |         |         |         |         |         |         |         |
21        |         |         |         |         |         |         |         |         |         |
22        |         |         |         |         |         |         |         |         |         |
23        |         |         |         |         |         |         |         |         |         |
24        |         |         |         |         |         |         |         |         |         |
25        |         |         |         |         |         |         |         |         |         |
26        |         |         |         |         |         |         |         |         |         |
27        |         |         |         |         |         |         |         |         |         |
28        |         |         |         |         |         |         |         |         |         |
29        |         |         |         |         |         |         |         |         |         |
30        |         |         |         |         |         |         |         |         |         |
31        |         |         |         |         |         |         |         |         |         |
32        |         |         |         |         |         |         |         |         |         |
33        |         |         |         |         |         |         |         |         |         |
34        |         |         |         |         |         |         |         |         |         |
35        |         |         |         |         |         |         |         |         |         |
36        |         |         |         |         |         |         |         |         |         |
37        |         |         |         |         |         |         |         |         |         |
38        |         |         |         |         |         |         |         |         |         |
39        |         |         |         |         |         |         |         |         |         |
40        |         |         |         |         |         |         |         |         |         |
41        |         |         |         |         |         |         |         |         |         |
42        |         |         |         |         |         |         |         |         |         |
43        |         |         |         |         |         |         |         |         |         |
44        |         |         |         |         |         |         |         |         |         |
45        |         |         |         |         |         |         |         |         |         |
46        |         |         |         |         |         |         |         |         |         |
47        |         |         |         |         |         |         |         |         |         |
48        |         |         |         |         |         |         |         |         |         |
49        |         |         |         |         |         |         |         |         |         |
50        |         |         |         |         |         |         |         |         |         |
51        |         |         |         |         |         |         |         |         |         |
52        |         |         |         |         |         |         |         |         |         |
53        |         |         |         |         |         |         |         |         |         |
54        |         |         |         |         |         |         |         |         |         |
55        |         |         |         |         |         |         |         |         |         |
56        |         |         |         |         |         |         |         |         |         |
57        |         |         |         |         |         |         |         |         |         |
58        |         |         |         |         |         |         |         |         |         |
59        |         |         |         |         |         |         |         |         |         |
60        |         |         |         |         |         |         |         |         |         |
61        |         |         |         |         |         |         |         |         |         |
62        |         |         |         |         |         |         |         |         |         |
63        |         |         |         |         |         |         |         |         |         |
64        |         |         |         |         |         |         |         |         |         |
65        |         |         |         |         |         |         |         |         |         |
66        |         |         |         |         |         |         |         |         |         |
67        |         |         |         |         |         |         |         |         |         |
68        |         |         |         |         |         |         |         |         |         |
69        |         |         |         |         |         |         |         |         |         |
70        |         |         |         |         |         |         |         |         |         |
71        |         |         |         |         |         |         |         |         |         |
72        |         |         |         |         |         |         |         |         |         |
73        |         |         |         |         |         |         |         |         |         |
74        |         |         |         |         |         |         |         |         |         |
75        |         |         |         |         |         |         |         |         |         |
76        |         |         |         |         |         |         |         |         |         |
77        |         |         |         |         |         |         |         |         |         |
78        |         |         |         |         |         |         |         |         |         |
79        |         |         |         |         |         |         |         |         |         |
80        |         |         |         |         |         |         |         |         |         |
81        |         |         |         |         |         |         |         |         |         |
82        |         |         |         |         |         |         |         |         |         |
83        |         |         |         |         |         |         |         |         |         |
84        |         |         |         |         |         |         |         |         |         |
85        |         |         |         |         |         |         |         |         |         |
86        |         |         |         |         |         |         |         |         |         |
87        |         |         |         |         |         |         |         |         |         |
88        |         |         |         |         |         |         |         |         |         |
89        |         |         |         |         |         |         |         |         |         |
90        |         |         |         |         |         |         |         |         |         |
91        |         |         |         |         |         |         |         |         |         |
92        |         |         |         |         |         |         |         |         |         |
93        |         |         |         |         |         |         |         |         |         |
94        |         |         |         |         |         |         |         |         |         |
95        |         |         |         |         |         |         |         |         |         |
96        |         |         |         |         |         |         |         |         |         |
97        |         |         |         |         |         |         |         |         |         |
98        |         |         |         |         |         |         |         |         |         |
99        |         |         |         |         |         |         |         |         |         |
100       |         |         |         |         |         |         |         |         |         |
101       |         |         |         |         |         |         |         |         |         |
102       |         |         |         |         |         |         |         |         |         |
103       |         |         |         |         |         |         |         |         |         |
104       |         |         |         |         |         |         |         |         |         |
105       |         |         |         |         |         |         |         |         |         |
106       |         |         |         |         |         |         |         |         |         |
107       |         |         |         |         |         |         |         |         |         |
108       |         |         |         |         |         |         |         |         |         |
109       |         |         |         |         |         |         |         |         |         |
110       |         |         |         |         |         |         |         |         |         |
111       |         |         |         |         |         |         |         |         |         |
112       |         |         |         |         |         |         |         |         |         |
113       |         |         |         |         |         |         |         |         |         |
114       |         |         |         |         |         |         |         |         |         |
115       |         |         |         |         |         |         |         |         |         |
116       |         |         |         |         |         |         |         |         |         |
117       |         |         |         |         |         |         |         |         |         |
118       |         |         |         |         |         |         |         |         |         |
119       |         |         |         |         |         |         |         |         |         |
120       |         |         |         |         |         |         |         |         |         |
121       |         |         |         |         |         |         |         |         |         |
122       |         |         |         |         |         |         |         |         |         |
123       |         |         |         |         |         |         |         |         |         |
124       |         |         |         |         |         |         |         |         |         |
125       |         |         |         |         |         |         |         |         |         |
126       |         |         |         |         |         |         |         |         |         |
127       |         |         |         |         |         |         |         |         |         |
128       |         |         |         |         |         |         |         |         |         |
129       |         |         |         |         |         |         |         |         |         |
130       |         |         |         |         |         |         |         |         |         |
131       |         |         |         |         |         |         |         |         |         |
132       |         |         |         |         |         |         |         |         |         |
133       |         |         |         |         |         |         |         |         |         |
134       |         |         |         |         |         |         |         |         |         |
135       |         |         |         |         |         |         |         |         |         |
136       |         |         |         |         |         |         |         |         |         |
137       |         |         |         |         |         |         |         |         |         |
138       |         |         |         |         |         |         |         |         |         |
139       |         |         |         |         |         |         |         |         |         |
140       |         |         |         |         |         |         |         |         |         |
141       |         |         |         |         |         |         |         |         |         |
142       |         |         |         |         |         |         |         |         |         |
143       |         |         |         |         |         |         |         |         |         |
144       |         |         |         |         |         |         |         |         |         |
145       |         |         |         |         |         |         |         |         |         |
146       |         |         |         |         |         |         |         |         |         |
147       |         |         |         |         |         |         |         |         |         |
148       |         |         |         |         |         |         |         |         |         |
149       |         |         |         |         |         |         |         |         |         |
150       |         |         |         |         |         |         |         |         |         |
151       |         |         |         |         |         |         |         |         |         |
152       |         |         |         |         |         |         |         |         |         |
153       |         |         |         |         |         |         |         |         |         |
154       |         |         |         |         |         |         |         |         |         |
155       |         |         |         |         |         |         |         |         |         |
156       |         |         |         |         |         |         |         |         |         |
157       |         |         |         |         |         |         |         |         |         |
158       |         |         |         |         |         |         |         |         |         |
159       |         |         |         |         |         |         |         |         |         |
160       |         |         |         |         |         |         |         |         |         |
161       |         |         |         |         |         |         |         |         |         |
162       |         |         |         |         |         |         |         |         |         |
163       |         |         |         |         |         |         |         |         |         |
164       |         |         |         |         |         |         |         |         |         |
165       |         |         |         |         |         |         |         |         |         |
166       |         |         |         |         |         |         |         |         |         |
167       |         |         |         |         |         |         |         |         |         |
168       |         |         |         |         |         |         |         |         |         |
169       |         |         |         |         |         |         |         |         |         |
170       |         |         |         |         |         |         |         |         |         |
171       |         |         |         |         |         |         |         |         |         |
172       |         |         |         |         |         |         |         |         |         |
173       |         |         |         |         |         |         |         |         |         |
174       |         |         |         |         |         |         |         |         |         |
175       |         |         |         |         |         |         |         |         |         |
176       |         |         |         |         |         |         |         |         |         |
177       |         |         |         |         |         |         |         |         |         |
178       |         |         |         |         |         |         |         |         |         |
179       |         |         |         |         |         |         |         |         |         |
180       |         |         |         |         |         |         |         |         |         |
181       |         |         |         |         |         |         |         |         |         |
182       |         |         |         |         |         |         |         |         |         |
183       |         |         |         |         |         |         |         |         |         |
184       |         |         |         |         |         |         |         |         |         |
185       |         |         |         |         |         |         |         |         |         |
186       |         |         |         |         |         |         |         |         |         |
187       |         |         |         |         |         |         |         |         |         |
188       |         |         |         |         |         |         |         |         |         |
189       |         |         |         |         |         |         |         |         |         |
190       |         |         |         |         |         |         |         |         |         |
191       |         |         |         |         |         |         |         |         |         |
192       |         |         |         |         |         |         |         |         |         |
193       |         |         |         |         |         |         |         |         |         |
194       |         |         |         |         |         |         |         |         |         |
195       |         |         |         |         |         |         |         |         |         |
196       |         |         |         |         |         |         |         |         |         |
197       |         |         |         |         |         |         |         |         |         |
198       |         |         |         |         |         |         |         |         |         |
199       |         |         |         |         |         |         |         |         |         |
200       |         |         |         |         |         |         |         |         |         |
201       |         |         |         |         |         |         |         |         |         |
202       |         |         |         |         |         |         |         |         |         |
203       |         |         |         |         |         |         |         |         |         |
204       |         |         |         |         |         |         |         |         |         |
205       |         |         |         |         |         |         |         |         |         |
206       |         |         |         |         |         |         |         |         |         |
207       |         |         |         |         |         |         |         |         |         |
208       |         |         |         |         |         |         |         |         |         |
209       |         |         |         |         |         |         |         |         |         |
210       |         |         |         |         |         |         |         |         |         |
211       |         |         |         |         |         |         |         |         |         |
212       |         |         |         |         |         |         |         |         |         |
213       |         |         |         |         |         |         |         |         |         |
214       |         |         |         |         |         |         |         |         |         |
215       |         |         |         |         |         |         |         |         |         |
216       |         |         |         |         |         |         |         |         |         |
217       |         |         |         |         |         |         |         |         |         |
218       |         |         |         |         |         |         |         |         |         |
219       |         |         |         |         |         |         |         |         |         |
220       |         |         |         |         |         |         |         |         |         |
221       |         |         |         |         |         |         |         |         |         |
222       |         |         |         |         |         |         |         |         |         |
223       |         |         |         |         |         |         |         |         |         |
224       |         |         |         |         |         |         |         |         |         |
225       |         |         |         |         |         |         |         |         |         |
226       |         |         |         |         |         |         |         |         |         |
227       |         |         |         |         |         |         |         |         |         |
228       |         |         |         |         |         |         |         |         |         |
229       |         |         |         |         |         |         |         |         |         |
230       |         |         |         |         |         |         |         |         |         |
231       |         |         |         |         |         |         |         |         |         |
232       |         |         |         |         |         |         |         |         |         |
233       |         |         |         |         |         |         |         |         |         |
234       |         |         |         |         |         |         |         |         |         |
235       |         |         |         |         |         |         |         |         |         |
236       |         |         |         |         |         |         |         |         |         |
237       |         |         |         |         |         |         |         |         |         |
238       |         |         |         |         |         |         |         |         |         |
239       |         |         |         |         |         |         |         |         |         |
240       |         |         |         |         |         |         |         |         |         |
241       |         |         |         |         |         |         |         |         |         |
242       |         |         |         |         |         |         |         |         |         |
243       |         |         |         |         |         |         |         |         |         |
244       |         |         |         |         |         |         |         |         |         |
245       |         |         |         |         |         |         |         |         |         |
246       |         |         |         |         |         |         |         |         |         |
247       |         |         |         |         |         |         |         |         |         |
248       |         |         |         |         |         |         |         |         |         |
249       |         |         |         |         |         |         |         |         |         |
250       |         |         |         |         |         |         |         |         |         |
251       |         |         |         |         |         |         |         |         |         |
252       |         |         |         |         |         |         |         |         |         |
253       |         |         |         |         |         |         |         |         |         |
254       |         |         |         |         |         |         |         |         |         |
255       |         |         |         |         |         |         |         |         |         |
256       |         |         |         |         |         |         |         |         |         |
257       |         |         |         |         |         |         |         |         |         |
258       |         |         |         |         |         |         |         |         |         |
259       |         |         |         |         |         |         |         |         |         |
260       |         |         |         |         |         |         |         |         |         |
261       |         |         |         |         |         |         |         |         |         |
262       |         |         |         |         |         |         |         |         |         |
263       |         |         |         |         |         |         |         |         |         |
264       |         |         |         |         |         |         |         |         |         |
265       |         |         |         |         |         |         |         |         |         |
266       |         |         |         |         |         |         |         |         |         |
267       |         |         |         |         |         |         |         |         |         |
268       |         |         |         |         |         |         |         |         |         |
269       |         |         |         |         |         |         |         |         |         |
270       |         |         |         |         |         |         |         |         |         |
271       |         |         |         |         |         |         |         |         |         |
272       |         |         |         |         |         |         |         |         |         |
273       |         |         |         |         |         |         |         |         |         |
274       |         |         |         |         |         |         |         |         |         |
275       |         |         |         |         |         |         |         |         |         |
276       |         |         |         |         |         |         |         |         |         |
277       |         |         |         |         |         |         |         |         |         |
278       |         |         |         |         |         |         |         |         |         |
279       |         |         |         |         |         |         |         |         |         |
280       |         |         |         |         |         |         |         |         |         |
281       |         |         |         |         |         |         |         |         |         |
282       |         |         |         |         |         |         |         |         |         |
283       |         |         |         |         |         |         |         |         |         |
284       |         |         |         |         |         |         |         |         |         |
285       |         |         |         |         |         |         |         |         |         |
286       |         |         |         |         |         |         |         |         |         |
287       |         |         |         |         |         |         |         |         |         |
288       |         |         |         |         |         |         |         |         |         |
289       |         |         |         |         |         |         |         |         |         |
290       |         |         |         |         |         |         |         |         |         |
291       |         |         |         |         |         |         |         |         |         |
292       |         |         |         |         |         |         |         |         |         |
293       |         |         |         |         |         |         |         |         |         |
294       |         |         |         |         |         |         |         |         |         |
295       |         |         |         |         |         |         |         |         |         |
296       |         |         |         |         |         |         |         |         |         |
297       |         |         |         |         |         |         |         |         |         |
298       |         |         |         |         |         |         |         |         |         |
299       |         |         |         |         |         |         |         |         |         |
300       |         |         |         |         |         |         |         |         |         |
301       |         |         |         |         |         |         |         |         |         |
302       |         |         |         |         |         |         |         |         |         |
303       |         |         |         |         |         |         |         |         |         |
304       |         |         |         |         |         |         |         |         |         |
305       |         |         |         |         |         |         |         |         |         |
306       |         |         |         |         |         |         |         |         |         |
307       |         |         |         |         |         |         |         |         |         |
308       |         |         |         |         |         |         |         |         |         |
309       |         |         |         |         |         |         |         |         |         |
310       |         |         |         |         |         |         |         |         |         |
311       |         |         |         |         |         |         |         |         |         |
312       |         |         |         |         |         |         |         |         |         |
313       |         |         |         |         |         |         |         |         |         |
314       |         |         |         |         |         |         |         |         |         |
315       |         |         |         |         |         |         |         |         |         |
316       |         |         |         |         |         |         |         |         |         |
317       |         |         |         |         |         |         |         |         |         |
318       |         |         |         |         |         |         |         |         |         |
319       |         |         |         |         |         |         |         |         |         |
320       |         |         |         |         |         |         |         |         |         |
321       |         |         |         |         |         |         |         |         |         |
322       |         |         |         |         |         |         |         |         |         |
323       |         |         |         |         |         |         |         |         |         |
324       |         |         |         |         |         |         |         |         |         |
325       |         |         |         |         |         |         |         |         |         |
326       |         |         |         |         |         |         |         |         |         |
327       |         |         |         |         |         |         |         |         |         |
328       |         |         |         |         |         |         |         |         |         |
329       |         |         |         |         |         |         |         |         |         |
330       |         |         |         |         |         |         |         |         |         |
331       |         |         |         |         |         |         |         |         |         |
332       |         |         |         |         |         |         |         |         |         |
333       |         |         |         |         |         |         |         |         |         |
334       |         |         |         |         |         |         |         |         |         |
335       |         |         |         |         |         |         |         |         |         |
336       |         |         |         |         |         |         |         |         |         |
337       |         |         |         |         |         |         |         |         |         |
338       |         |         |         |         |         |         |         |         |         |
339       |         |         |         |         |         |         |         |         |         |
340       |         |         |         |         |         |         |         |         |         |
341       |         |         |         |         |         |         |         |         |         |
342       |         |         |         |         |         |         |         |         |         |
343       |         |         |         |         |         |         |         |         |         |
344       |         |         |         |         |         |         |         |         |         |
345       |         |         |         |         |         |         |         |         |         |
346       |         |         |         |         |         |         |         |         |         |
347       |         |         |         |         |         |         |         |         |         |
348       |         |         |         |         |         |         |         |         |         |
349       |         |         |         |         |         |         |         |         |         |
350       |         |         |         |         |         |         |         |         |         |
351       |         |         |         |         |         |         |         |         |         |
352       |         |         |         |         |         |         |         |         |         |
353       |         |         |         |         |         |         |         |         |         |
354       |         |         |         |         |         |         |         |         |         |
355       |         |         |         |         |         |         |         |         |         |
356       |         |         |         |         |         |         |         |         |         |
357       |         |         |         |         |         |         |         |         |         |
358       |         |         |         |         |         |         |         |         |         |
359       |         |         |         |         |         |         |         |         |         |
360       |         |         |         |         |         |         |         |         |         |
361       |         |         |         |         |         |         |         |         |         |
362       |         |         |         |         |         |         |         |         |         |
363       |         |         |         |         |         |         |         |         |         |
364       |         |         |         |         |         |         |         |         |         |
365       |         |         |         |         |         |         |         |         |         |
366       |         |         |         |         |         |         |         |         |         |
367       |         |         |         |         |         |         |         |         |         |
368       |         |         |         |         |         |         |         |         |         |
369       |         |         |         |         |         |         |         |         |         |
370       |         |         |         |         |         |         |         |         |         |
371       |         |         |         |         |         |         |         |         |         |
372       |         |         |         |         |         |         |         |         |         |
373       |         |         |         |         |         |         |         |         |         |
374       |         |         |         |         |         |         |         |         |         |
375       |         |         |         |         |         |         |         |         |         |
376       |         |         |         |         |         |         |         |         |         |
377       |         |         |         |         |         |         |         |         |         |
378       |         |         |         |         |         |         |         |         |         |
379       |         |         |         |         |         |         |         |         |         |
380       |         |         |         |         |         |         |         |         |         |
381       |         |         |         |         |         |         |         |         |         |
382       |         |         |         |         |         |         |         |         |         |
383       |         |         |         |         |         |         |         |         |         |
384       |         |         |         |         |         |         |         |         |         |
385       |         |         |         |         |         |         |         |         |         |
386       |         |         |         |         |         |         |         |         |         |
387       |         |         |         |         |         |         |         |         |         |
388       |         |         |         |         |         |         |         |         |         |
389       |         |         |         |         |         |         |         |         |         |
390       |         |         |         |         |         |         |         |         |         |
391       |         |         |         |         |         |         |         |         |         |
392       |         |         |         |         |         |         |         |         |         |
393       |         |         |         |         |         |         |         |         |         |
394       |         |         |         |         |         |         |         |         |         |
395       |         |         |         |         |         |         |         |         |         |
396       |         |         |         |         |         |         |         |         |         |
397       |         |         |         |         |         |         |         |         |         |
398       |         |         |         |         |         |         |         |         |         |
399       |         |         |         |         |         |         |         |         |         |
400       |         |         |         |         |         |         |         |         |         |
401       |         |         |         |         |         |         |         |         |         |
402       |         |         |         |         |         |         |         |         |         |
403       |         |         |         |         |         |         |         |         |         |
404       |         |         |         |         |         |         |         |         |         |
405       |         |         |         |         |         |         |         |         |         |
406       |         |         |         |         |         |         |         |         |         |
407       |         |         |         |         |         |         |         |         |         |
408       |         |         |         |         |         |         |         |         |         |
409       |         |         |         |         |         |         |         |         |         |
410       |         |         |         |         |         |         |         |         |         |
411       |         |         |         |         |         |         |         |         |         |
412       |         |         |         |         |         |         |         |         |         |
413       |         |         |         |         |         |         |         |         |         |
414       |         |         |         |         |         |         |         |         |         |
415       |         |         |         |         |         |         |         |         |         |
416       |         |         |         |         |         |         |         |         |         |
417       |         |         |         |         |         |         |         |         |         |
418       |         |         |         |         |         |         |         |         |         |
419       |         |         |         |         |         |         |         |         |         |
420       |         |         |         |         |         |         |         |         |         |
421       |         |         |         |         |         |         |         |         |         |
422       |         |         |         |         |         |         |         |         |         |
423       |         |         |         |         |         |         |         |         |         |
424       |         |         |         |         |         |         |         |         |         |
425       |         |         |         |         |         |         |         |         |         |
426       |         |         |         |         |         |         |         |         |         |
427       |         |         |         |         |         |         |         |         |         |
428       |         |         |         |         |         |         |         |         |         |
429       |         |         |         |         |         |         |         |         |         |
430       |         |         |         |         |         |         |         |         |         |
431       |         |         |         |         |         |         |         |         |         |
432       |         |         |         |         |         |         |         |         |         |
433       |         |         |         |         |         |         |         |         |         |
434       |         |         |         |         |         |         |         |         |         |
435       |         |         |         |         |         |         |         |         |         |
436       |         |         |         |         |         |         |         |         |         |
437       |         |         |         |         |         |         |         |         |         |
438       |         |         |         |         |         |         |         |         |         |
439       |         |         |         |         |         |         |         |         |         |
440       |         |         |         |         |         |         |         |         |         |
441       |         |         |         |         |         |         |         |         |         |
442       |         |         |         |         |         |         |         |         |         |
443       |         |         |         |         |         |         |         |         |         |
444       |         |         |         |         |         |         |         |         |         |
445       |         |         |         |         |         |         |         |         |         |
446       |         |         |         |         |         |         |         |         |         |
447       |         |         |         |         |         |         |         |         |         |
448       |         |         |         |         |         |         |         |         |         |
449       |         |         |         |         |         |         |         |         |         |
450       |         |         |         |         |         |         |         |         |         |
451       |         |         |         |         |         |         |         |         |         |
452       |         |         |         |         |         |         |         |         |         |
453       |         |         |         |         |         |         |         |         |         |
454       |         |         |         |         |         |         |         |         |         |
455       |         |         |         |         |         |         |         |         |         |
456       |         |         |         |         |         |         |         |         |         |
457       |         |         |         |         |         |         |         |         |         |
458       |         |         |         |         |         |         |         |         |         |
459       |         |         |         |         |         |         |         |         |         |
460       |         |         |         |         |         |         |         |         |         |
461       |         |         |         |         |         |         |         |         |         |
462       |         |         |         |         |         |         |         |         |         |
463       |         |         |         |         |         |         |         |         |         |
464       |         |         |         |         |         |         |         |         |         |
465       |         |         |         |         |         |         |         |         |         |
466       |         |         |         |         |         |         |         |         |         |
467       |         |         |         |         |         |         |         |         |         |
468       |         |         |         |         |         |         |         |         |         |
469       |         |         |         |         |         |         |         |         |         |
470       |         |         |         |         |         |         |         |         |         |
471       |         |         |         |         |         |         |         |         |         |
472       |         |         |         |         |         |         |         |         |         |
473       |         |         |         |         |         |         |         |         |         |
474       |         |         |         |         |         |         |         |         |         |
475       |         |         |         |         |         |         |         |         |         |
476       |         |         |         |         |         |         |         |         |         |
477       |         |         |         |         |         |         |         |         |         |
478       |         |         |         |         |         |         |         |         |         |
479       |         |         |         |         |         |         |         |         |         |
480       |         |         |         |         |         |         |         |         |         |
481       |         |         |         |         |         |         |         |         |         |
482       |         |         |         |         |         |         |         |         |         |
483       |         |         |         |         |         |         |         |         |         |
484       |         |         |         |         |         |         |         |         |         |
485       |         |         |         |         |         |         |         |         |         |
486       |         |         |         |         |         |         |         |         |         |
487       |         |         |         |         |         |         |         |         |         |
488       |         |         |         |         |         |
```

A medida "Bugs All" apresentada na Figura 4.8 tem como objetivo contabilizar o número total de *bugs* reportados em ambientes críticos, especificamente em *staging* e *production*, com base nos registos da tabela 'WorkItems (2)'. Esta métrica é essencial para equipas de qualidade e desenvolvimento, pois permite uma visão consolidada de problemas em ambientes que precedem ou representam o cenário real de utilização do *software*.

A medida utiliza uma estrutura típica de DAX, começando pela declaração de uma variável *bugsstag*, que armazena o resultado do cálculo intermédio. Esta variável emprega a função *CALCULATE* para modificar o contexto de filtro e contar as linhas da tabela 'WorkItems (2)' que satisfazem determinadas condições. A função *COUNTROWS* é utilizada para contar o número total de registos, enquanto a condição aplicada verifica se a coluna 'Custom_Environment' contém a palavra "Staging" ou "Production". Esta verificação é feita através da combinação lógica // (OR) entre duas funções *CONTAINSSTRING*, garantindo que qualquer registo que mencione um destes ambientes seja incluído na contagem.

A segunda parte da medida, iniciada pela instrução *RETURN*, assegura que o resultado final seja sempre um valor numérico válido, mesmo que não existam *bugs* registados nos ambientes especificados. A função *IF* combinada com *ISBLANK* verifica se a variável *bugsstag* está vazia e, em caso afirmativo, retorna 0, evitando assim a exibição de valores nulos em relatórios. Caso contrário, o valor calculado em *bugsstag* é devolvido.

Esta medida destaca-se pela sua abrangência, permitindo monitorizar bugs em múltiplos ambientes com uma única expressão. Em resumo, a medida "Bugs All" oferece uma visão unificada de problemas em ambientes críticos, facilitando a identificação de padrões e a priorização de correções. Ao agregar *bugs* de *staging* e *production*, esta métrica ajuda as equipas a antecipar potenciais problemas antes que estes afetem os utilizadores finais, contribuindo para a melhoria contínua da qualidade e estabilidade do *software*.

```

1 Bugs All =
2     VAR bugsstag = CALCULATE(
3         COUNTROWS('WorkItems (2)'),CONTAINSSTRING('WorkItems (2)'[Custom_Environment],
4             "Staging") || CONTAINSSTRING('WorkItems (2)'[Custom_Environment],"Production")
5     )
6     RETURN IF( ISBLANK(bugsstag), 0, bugsstag)

```

Figura 4.8. Expressão DAX para contabilizar o número total de bugs reportados em ambientes críticos

DevOps and (Software) Operational Excellence Metrics

A medida "*Bugs Staging*" apresentada na Figura 4.9 utiliza uma abordagem inteligente e eficiente para calcular o número de *bugs* específicos do ambiente de *staging*, demonstrando como medidas DAX podem ser combinadas para criar métricas derivadas de forma simples, mas poderosa.

Esta medida opera através de uma subtração direta entre duas outras medidas previamente definidas: "*[Bugs All]*" (que conta *bugs* em *staging* e *production*) e "*[Bugs Production]*" (que conta apenas *bugs* em *production*). O resultado desta operação matemática revela naturalmente a quantidade de *bugs* existentes exclusivamente no ambiente de *staging*, sem necessidade de repetir filtros ou condições já aplicados nas medidas originais.

A simplicidade desta solução é enganadora - por trás desta única linha de código está um conceito importante de modelagem de dados: o reaproveitamento de medidas existentes. Esta prática não só reduz a complexidade do modelo como garante consistência nos cálculos, já que qualquer ajuste nas medidas base (*[Bugs All]* ou *[Bugs Production]*) automaticamente se reflete nesta medida derivada.

Vale destacar que esta abordagem assume implicitamente que a medida *[Bugs All]* contém exatamente a soma de *bugs* de *staging* e *production*, sem outros ambientes incluídos. Esta premissa é crucial para a precisão do cálculo e reflete uma decisão consciente de design do modelo de dados. Caso novos ambientes venham a ser incluídos na medida *[Bugs All]*, seria necessário rever esta lógica.

A medida não inclui tratamento explícito para valores nulos ou negativos, o que é aceitável neste contexto pois as medidas base já incorporam essa lógica (como visto nos exemplos anteriores com *ISBLANK*). Se porventura *[Bugs Production]* for maior que *[Bugs All]*, o resultado negativo poderia indicar algum problema nos dados subjacentes que mereceria investigação.

Esta solução elegante exemplifica como o DAX permite criar hierarquias de medidas, onde métricas complexas podem ser construídas a partir de componentes mais simples, facilitando a manutenção e melhorando a clareza do modelo analítico como um todo.

1 Bugs Staging = [Bugs All] - [Bugs Production]

Figura 4.9. Fórmula DAX para calcular o número de bugs específicos do ambiente de staging

A medida "Defect Escape Rate %" apresentada na Figura 4.10 calcula uma métrica crítica de qualidade em processos de desenvolvimento de *software*: a taxa de defeitos que "escapam" para o ambiente de produção, expressa como uma percentagem. Este indicador é fundamental para avaliar a eficácia dos processos de teste e controlo de qualidade antes da implementação em produção.

A estrutura da medida utiliza a abordagem padrão em DAX com variáveis e tratamento de valores nulos. A variável *rate* é definida através da função *DIVIDE*, que realiza uma divisão segura entre duas medidas existentes: *WorkItems (2)[Bugs Production]* (que conta os *bugs* encontrados em produção) e *WorkItems (2)[Bugs All]* (que representa o total de *bugs* identificados em todos os ambientes monitorizados). O terceiro argumento da função *DIVIDE* (0) especifica o valor padrão a retornar caso ocorra uma divisão por zero, prevenindo erros. O resultado da divisão é então multiplicado por 100 para converter o valor em percentagem.

A segunda parte da medida, iniciada por *RETURN*, aplica uma camada adicional de proteção contra valores nulos através da combinação *IF(ISBLANK(rate), 0, rate)*. Esta construção garante que, mesmo em situações excecionais onde a função *DIVIDE* pudesse retornar um valor nulo (algo teoricamente impossível neste caso específico, dado o 0 como valor padrão), a medida sempre retornaria um valor numérico válido (0) em vez de um valor nulo que poderia causar problemas em visualizações ou cálculos subsequentes.

Esta medida revela uma importante característica de boas práticas em DAX: a defesa em profundidade. Apesar da função *DIVIDE* já incluir tratamento para divisões inválidas, a medida adiciona ainda uma verificação adicional para valores nulos, demonstrando uma abordagem cautelosa que é particularmente importante em métricas críticas como esta. O cálculo em si é simples, mas importante - ao dividir *bugs* em produção pelo total de *bugs* identificados, obtém-se imediatamente uma visão clara da proporção de problemas que não foram detetados durante os estágios anteriores do ciclo de desenvolvimento.

Vale notar que esta métrica assume que *'WorkItems (2) [Bugs All]* inclui pelo menos todos os *bugs* contados em *'WorkItems (2) [Bugs Production]*, uma premissa que deve ser validada na estrutura geral do modelo de dados. Um valor próximo de 0% indicaria um processo de teste muito eficaz, enquanto valores mais elevados sugeririam oportunidades para melhorar os processos de controlo de qualidade antes da implementação em produção. A apresentação do resultado em percentagem facilita a interpretação e comparação ao longo do tempo ou entre diferentes equipas ou produtos.

```
1 Defect Escape Rate % =  
2 VAR rate = DIVIDE('WorkItems (2) [Bugs Production]', 'WorkItems (2) [Bugs All]', 0)*100  
3 RETURN IF(ISBLANK(rate), 0, rate)
```

Figura 4.10. Fórmula DAX para Taxa de Defeitos em Produção (%)

4.3.2.4 Criação do dashboard em Power BI

O *Power BI* permite criar *dashboards* para visualizar dados de forma interativa, nos quais os elementos visuais podem interagir através de filtros cruzados. Cada *dashboard* oferece um conjunto de características que conferem ao utilizador maior liberdade de visualização e a possibilidade de alterar cenários.

Os *dashboards* em questão foram meticulosamente concebidos para proporcionar uma experiência de utilização dinâmica e intuitiva, permitindo aos utilizadores filtrar, analisar e visualizar dados de forma eficiente, consoante as suas necessidades específicas. A arquitetura assenta em duas dimensões fundamentais de filtragem — temporal e organizacional — complementadas por mecanismos inteligentes de gestão de espaço, como os botões de exibição e ocultação de tabelas.

Os filtros temporais (ver Figura 4.11) oferecem uma granularidade detalhada, permitindo segmentar os dados em múltiplos níveis hierárquicos, desde amplos períodos anuais até a dias específicos. Esta estrutura facilita tanto análises macro, como tendências de longo prazo, como micro, focadas em pormenores diários:

- Ano: Seleção de um ano específico para análise transversal.
- Trimestre: Filtragem por trimestres (Q1 a Q4), ideal para avaliações periódicas de desempenho.

DevOps and (Software) Operational Excellence Metrics

- Mês: Permite focar em meses particulares, útil para identificar sazonalidades ou anomalias.
- Dia: Máximo detalhe para análises pontuais ou revisão de eventos específicos.

A opção "Seleções múltiplas" (*Multiple selections*) amplia a flexibilidade, possibilitando, por exemplo, comparar dados entre vários trimestres ou meses distintos numa única visualização, sem necessidade de reconfigurações repetidas.

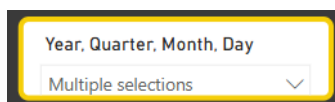


Figura 4.11. Filtros Temporais: análise com precisão cronológica

Para além da dimensão temporal, os *dashboards* permite filtrar dados com base na estrutura organizacional da empresa (Figura 4.12), adaptando-se a diferentes níveis de gestão:

- Área: Agrupamento por grandes divisões funcionais (ex.: Tecnologia, Operações).
- Departamento: Segmentação mais fina dentro de cada área (ex.: Desenvolvimento, Infraestruturas).
- Equipa: Foco em grupos de trabalho específicos.
- Projeto: Filtragem por iniciativas ou tarefas em curso.

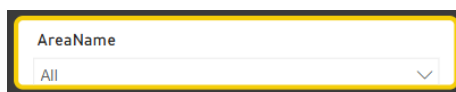


Figura 4.12. Filtros Organizacionais: segmentação por estrutura empresarial

A lista de áreas disponíveis (como ilustrado na Figura 4.13) abrange um espectro alargado, incluindo, entre outras:

- *Anova Connect* e *Anova Go* (soluções tecnológicas).
- *Backend Team* e *FrontEnd Enabling Team* (equipas de desenvolvimento).
- *DevOps Approvers* e *Infra and Data* (infraestruturas e gestão de dados).
- *Design* e *Consumer Mobile App* (áreas criativas e de produto).

DevOps and (Software) Operational Excellence Metrics

Estes filtros são indispensáveis para gestores que necessitem de monitorizar o desempenho de equipas ou projetos específicos, eliminando ruído informativo e concentrando-se apenas nos dados relevantes.

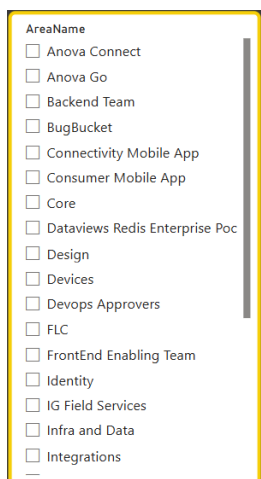


Figura 4.13. Lista de áreas disponíveis

Em *dashboards* com elevada densidade informativa, a gestão do espaço é crítica. Para tal, foram implementados botões de ação rápida do tipo marcador, com as funções "Mostrar Tabela" (*Show Table*) e "Ocultar Tabela" (*Hide Table*). Estas funcionalidades (Figura 4.14) permitem:

- Exibir tabelas: Quando o utilizador necessita de aceder a dados brutos ou detalhados em formato tabular, bastando um clique para expandir a informação.
- Ocultar tabelas: Para simplificar a visualização em ecrãs mais compactos ou privilegiar gráficos e métricas sumárias, as tabelas podem ser recolhidas, garantindo uma experiência mais limpa e focada.

Esta abordagem adaptativa é particularmente útil em dispositivos com ecrãs reduzidos ou em apresentações onde a clareza visual é prioritária.



Figura 4.14. Botões de ação rápida do tipo marcador

DevOps and (Software) Operational Excellence Metrics

Os *dashboards* foram pensados para se adaptar a diferentes perfis de utilizadores, desde analistas de dados a gestores de equipa. As suas funcionalidades permitem:

- Combinação de filtros: Cruzar critérios temporais e organizacionais (ex.: dados do "Core" no 2.º trimestre de 2024).
- Interatividade: Atualização em tempo real dos gráficos e tabelas consoante os filtros aplicados.
- Economia de espaço: Alternância fluida entre modos de visualização para evitar sobrecarga informativa.

Em síntese, estes *dashboards* destacam-se pela sua flexibilidade, precisão e capacidade de adaptação às necessidades dos utilizadores. A combinação de filtros temporais e organizacionais avançados, aliada a mecanismos inteligentes de gestão de espaço, cria uma ferramenta robusta para análise de dados e tomada de decisões informadas.

Seja para identificar tendências anuais, comparar o desempenho de equipas ou focar em eventos diários, esta solução oferece um equilíbrio perfeito entre detalhe e simplicidade, tornando-se um recurso indispensável em contextos empresariais onde a clareza e a eficiência são prioritárias.

Com os pormenores criados e parametrizados, no subcapítulo 4.4 procede-se à ilustração dos vários *dashboards* criados, bem como as visualizações utilizadas.

4.4 Demonstração

De seguida, apresentam-se os *dashboards* de métricas *DevOps* desenvolvidos em *Power BI*, concebidos para monitorizar e analisar métricas relacionadas com implementações, qualidade de *software* e eficiência de processos. Demonstra-se de que forma o sistema de BI criado resolve o problema inicial de falta de visibilidade e de análise consolidada destes indicadores.

Foi desenvolvido um conjunto de quatro *dashboards* independentes, com uma estrutura semelhante, cada um dedicado a uma métrica fundamental do ciclo de *DevOps*. Foram incluídos filtros interativos que permitem segmentar os dados por equipa, período ou projeto. Por exemplo, o utilizador pode selecionar um trimestre específico para analisar a

performance de uma equipa ou comparar meses distintos. A implementação de filtros personalizados e a diversidade de visualizações garantem que utilizadores com diferentes perfis — desde *developers* a gestores — possam interpretar os dados de forma eficaz.

Para garantir uma experiência intuitiva, todas as visualizações incluem *tooltips* que fornecem informações adicionais quando o utilizador passa o cursor sobre os dados. Por exemplo, ao visualizar o gráfico do número de implementações por mês, é possível ver detalhes como o número exato de *deploys* num mês específico.

O primeiro *dashboard* (Figura 4.15), dedicado à Frequência de Implementações (*Deployment Frequency*), foi concebido para monitorizar a cadência de *releases*, elemento crítico para avaliar a maturidade dos processos de integração e entrega contínua. Neste painel, os indicadores principais incluem: o volume total de implementações, a média diária de *deploys*, a distribuição temporal (diária e mensal) e a identificação de padrões sazonais. A visualização combina gráficos de colunas agrupadas para mostrar a evolução temporal, complementados por indicadores numéricos que destacam os valores mais recentes nos últimos 28 dias.

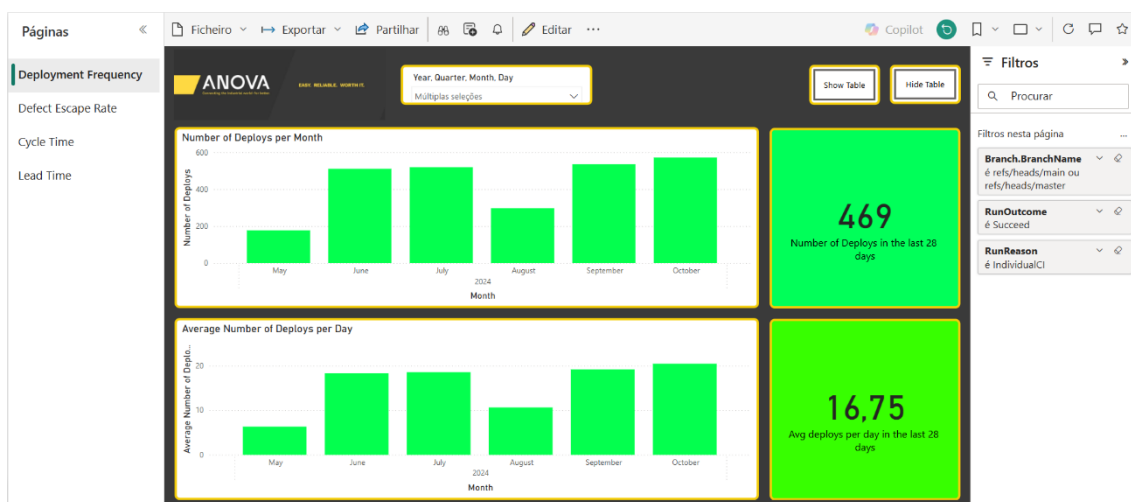


Figura 4.15. Dashboard da Frequência de Implementações (*Deployment Frequency*)

O segundo *dashboard* (Figura 4.16), focado na Taxa de Defeitos em Produção (*Defect Escape Rate*), foi estruturado para avaliar a eficácia dos processos de controlo de qualidade. Aqui, a análise centra-se na percentagem de erros que atingem ambiente produtivo, um indicador crucial da robustez dos testes automatizados e da revisão de código. Este painel

DevOps and (Software) Operational Excellence Metrics

incorpora: uma série temporal da taxa mensal de defeitos em produção, um *breakdown* da origem dos problemas (produção vs. ambientes não produtivos), e uma análise comparativa entre equipas ou componentes do sistema. A representação gráfica utiliza um gráfico de colunas agrupadas para a evolução histórica, complementado por um gráfico circular para mostrar a proporção atual de *bugs* em diferentes ambientes, detalhando a natureza e severidade de cada incidente.



Figura 4.16. Dashboard da Taxa de Defeitos em Produção (Defect Escape Rate)

Para a visualização das métricas mencionadas anteriormente, foram utilizados gráficos de colunas agrupadas e cartões. Inclui-se ainda um gráfico circular que apresenta o número e a percentagem de *bugs* em *production* e em *staging* nos últimos 28 dias.

Além das métricas já referidas e dos respetivos elementos visuais associados, os *dashboards* contêm filtros que permitem segmentar os dados por período temporal (ano, trimestre, mês ou dia) e por área (área, departamento, equipa ou projeto). Incluem também tabelas que possibilitam uma consulta organizada e detalhada dos dados mais relevantes.

Para garantir uma visualização mais clara e devido a limitações de espaço, foram implementados botões de tipo marcador em algumas páginas dos *dashboards*, com a funcionalidade de exibir ou ocultar tabelas conforme necessário. De seguida, serão apresentadas as tabelas ocultas relativas ao primeiro e segundo *dashboards* (Figura 4.17 e Figura 4.18).

DevOps and (Software) Operational Excellence Metrics

Year	Quarter	Month	Day	RunReason	RunOutcome	Number of Deploys
2024	Qtr 2	May	1	IndividualCI	Succeed	1
2024	Qtr 2	May	2	IndividualCI	Succeed	6
2024	Qtr 2	May	3	IndividualCI	Succeed	7
2024	Qtr 2	May	4	IndividualCI	Succeed	0
2024	Qtr 2	May	5	IndividualCI	Succeed	0
2024	Qtr 2	May	6	IndividualCI	Succeed	7
2024	Qtr 2	May	7	IndividualCI	Succeed	7
2024	Qtr 2	May	8	IndividualCI	Succeed	7
2024	Qtr 2	May	9	IndividualCI	Succeed	14
2024	Qtr 2	May	10	IndividualCI	Succeed	5
2024	Qtr 2	May	11	IndividualCI	Succeed	0
Total						2612

Figura 4.17. Tabela oculta presente no Dashboard da Frequência de Implementações (Deployment Frequency)

Year	Quarter	Month	Day	WorkItemType	Severity	Custom_Environment
2023	Qtr 1	January	2	Bug	3 - Medium	Production
2023	Qtr 1	January	2	Bug	3 - Medium	Staging
2023	Qtr 1	January	3	Bug	1 - Critical	Production
2023	Qtr 1	January	3	Bug	3 - Medium	Staging
2023	Qtr 1	January	4	Bug	3 - Medium	Production
2023	Qtr 1	January	4	Bug	3 - Medium	Staging
2023	Qtr 1	January	5	Bug	2 - High	Production
2023	Qtr 1	January	5	Bug	3 - Medium	Staging
2023	Qtr 1	January	6	Bug	3 - Medium	Staging
2023	Qtr 1	January	9	Bug	3 - Medium	Production
2023	Qtr 1	January	9	Bug	3 - Medium	Staging
2023	Qtr 1	January	9	Bug	4 - Low	Staging

Figura 4.18. Tabela oculta presente no Dashboard da Taxa de Defeitos em Produção (Defect Escape Rate)

O terceiro (Figura 4.19), dedicado ao Tempo de Ciclo (Cycle Time), foi desenhado para medir a eficiência do fluxo de trabalho de desenvolvimento. Este dashboard apresenta uma análise multidimensional que inclui: a média histórica do tempo necessário para concluir tarefas desde o commit até à implementação, e a identificação de outliers que possam indicar bloqueios processuais. A visualização principal utiliza tabelas, gráficos de colunas agrupadas e cartões, permitindo identificar tanto tendências de longo prazo como variações pontuais.

DevOps and (Software) Operational Excellence Metrics

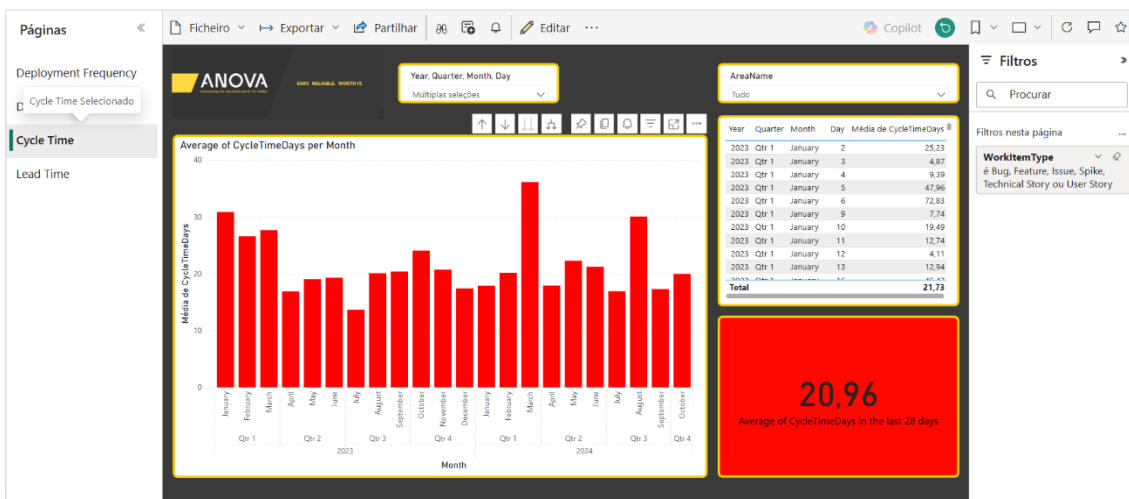


Figura 4.19. Dashboard do Tempo de Ciclo (Cycle Time)

O quarto e último dashboard (Figura 4.20), centrado no Tempo de Espera (Lead Time), foi concebido para analisar o tempo total desde a criação do ticket até à resolução efetiva em produção, incluindo períodos de espera e processamento. Este painel diferencia-se por focar-se nos tempos totais do processo, com indicadores que incluem: o lead time médio por tipo de tarefa, a comparação entre equipas ou projetos, e a análise dos componentes do tempo total (tempo ativo vs. tempo em espera). A representação visual utiliza tabelas, gráficos de colunas agrupadas e cartões para decompor e identificar padrões temporais e gargalos processuais.

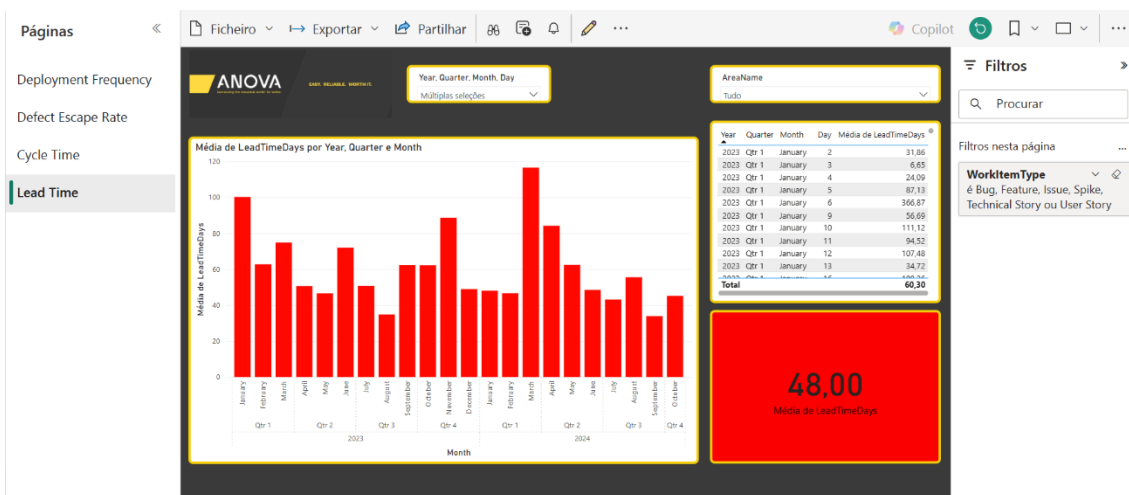


Figura 4.20. Dashboard do Tempo de Espera (Lead Time)

4.5 Avaliação

Com o projeto na fase de finalização, realizou-se outra reunião com o *Chief Software Architect* da ANOVA para apresentar a solução desenvolvida e avaliar o trabalho realizado. Neste encontro, além de se destacar a solução inicialmente proposta, discutiram-se sugestões de alteração e apresentaram-se as novidades implementadas em resposta ao *feedback* recebido. Dado que, ao longo do estágio, se realizaram várias reuniões de acompanhamento, nesta fase final não foram necessárias alterações de fundo nas métricas, dados, componentes visuais ou estrutura do relatório. Concluída a reunião, verificou-se que as alterações necessárias se circunscreviam à formatação condicional das visualizações.

No âmbito da revisão de literatura efetuada durante o estágio, constatou-se que o verdadeiro valor das métricas *DevOps* reside não na sua aplicação genérica, mas na capacidade de as ajustar dinamicamente, garantindo que refletem as prioridades e realidades específicas de cada organização. Esta abordagem contextualizada aliada a uma avaliação contínua é essencial para evitar conclusões enganadoras ou incentivos distorcidos. A flexibilidade na seleção e interpretação das métricas constitui, portanto, um elemento fundamental para evitar análises distorcidas ou incentivos contraproducentes.

Conclui-se, portanto, que, em vez de uma adoção indiscriminada de métricas universais, as organizações devem ajustá-las continuamente, alinhando-as com os seus processos e objetivos específicos, de modo a obter informações verdadeiramente úteis e acionáveis.

Com base na revisão da literatura, nomeadamente no relatório DORA (2024), nas recomendações e melhores práticas da área de *DevOps*, e tendo em consideração o contexto, os processos e os objetivos específicos da ANOVA, foram definidas as classificações para as métricas, bem como a respetiva formatação condicional dos visuais associados. Com o apoio e validação do supervisor de estágio, estabeleceram-se as seguintes categorias:

- **Elite** (representada a verde nos visuais): Organizações com desempenho de elite caracterizam-se por uma elevada eficiência, processos altamente automatizados e maduros, capacidade de implementar alterações com rapidez e uma incidência muito reduzida de falhas em produção.

DevOps and (Software) Operational Excellence Metrics

- **High** (representada a azul nos visuais): Organizações de alto desempenho apresentam processos ágeis e eficazes, embora ainda exista margem para melhorias ao nível da automatização e da qualidade das entregas.
- **Medium** (representada a amarelo nos visuais): Organizações com desempenho médio revelam processos moderados, enfrentando desafios relevantes em termos de velocidade de entrega e fiabilidade operacional.
- **Low** (representada a vermelho nos visuais): Organizações com desempenho baixo possuem processos lentos, maioritariamente manuais e suscetíveis a erros, exigindo intervenções significativas para melhorar a agilidade e a fiabilidade.

Para uma melhor leitura dos KPI e respetivas categorias de valores criaram-se três tabelas, uma com a lista de dashboards implementados (Tabela 4.2), outra com os KPI implementados e dashboards a que pertencem (

Tabela 4.3) e uma com as categorias de performance dos KPI (Tabela 4.4).

DevOps and (Software) Operational Excellence Metrics

Tabela 4.2. ID e designação dos dashboards

Dashboard ID	Dashboard Designação
DASH 1	Dashboard da Frequência de Implementações (<i>Deployment Frequency</i>)
DASH 2	Dashboard da Taxa de Defeitos em Produção (<i>Defect Escape Rate</i>)
DASH 3	Dashboard do Tempo de Ciclo (<i>Cycle Time</i>)
DASH 4	Dashboard do Tempo de Espera (<i>Lead Time</i>)

Tabela 4.3. ID, designação e identificação dos dashboards a que pertencem os KPI

KPI ID	KPI Designação	Dashboard a que pertence o KPI
KPI 1	<i>Number of Deploys per Month</i> (Número de Implementações por Mês)	DASH 1
KPI 2	<i>Average Number of Deploys per Day</i> (Média de Implementações por Dia)	DASH 1
KPI 3	<i>Number of Deploys in the last 28 days</i> (Número de Implementações nos últimos 28 dias)	DASH 1
KPI 4	<i>Average Deploys per Day in the last 28 days</i> (Média de Implementações por Dia nos últimos 28 dias)	DASH 1
KPI 5	<i>Defect Escape Rate per Month</i> (Taxa de Defeitos que "Escapam" por Mês)	DASH 2
KPI 6	<i>Defect Escape Rate in the last 28 days</i> (Taxa de Defeitos que "Escapam" nos últimos 28 dias)	DASH 2
KPI 7	<i>Average of Cycle Time Days per Month</i> (Média de Dias de Tempo de Ciclo por Mês)	DASH 3
KPI 8	<i>Average of Cycle Time Days in the last 28 days</i> (Média de Dias de Tempo de Ciclo nos últimos 28 dias)	DASH 3
KPI 9	<i>Average of Lead Time Days per Month</i> (Média de Dias de Tempo de Espera por Mês)	DASH 4
KPI 10	<i>Average of Lead Time Days in the last 28 days</i> (Média de Dias de Tempo de Espera nos últimos 28 dias)	DASH 4

DevOps and (Software) Operational Excellence Metrics

Tabela 4.4. Categorias de performance dos KPI

KPI/Nível de Performance	Elite	High	Medium	Low
KPI 1	> 100	30-100	10-30	< 10
KPI 2	> 5	1-5	0,5-1	< 0,5
KPI 3	> 100	30-100	10-30	< 10
KPI 4	> 5	1-5	0,5-1	< 0,5
KPI 5	0%-5%	6%-15%	16%-30%	> 30%
KPI 6	0%-5%	6%-15%	16%-30%	> 30%
KPI 7	< 1	1 a 7	7 a 30	> 30
KPI 8	< 1	1 a 7	7 a 30	> 30
KPI 9	< 7	7 a 30	30 a 90	> 90
KPI 10	< 7	7 a 30	30 a 90	> 90

4.5.1 Dashboards finais

As figuras seguintes apresentam-se os dashboards implementados e respetivos KPI após a avaliação dos mesmos por parte da ANOVA.

- *Dashboard da Frequência de Implementações (DASH 1)*

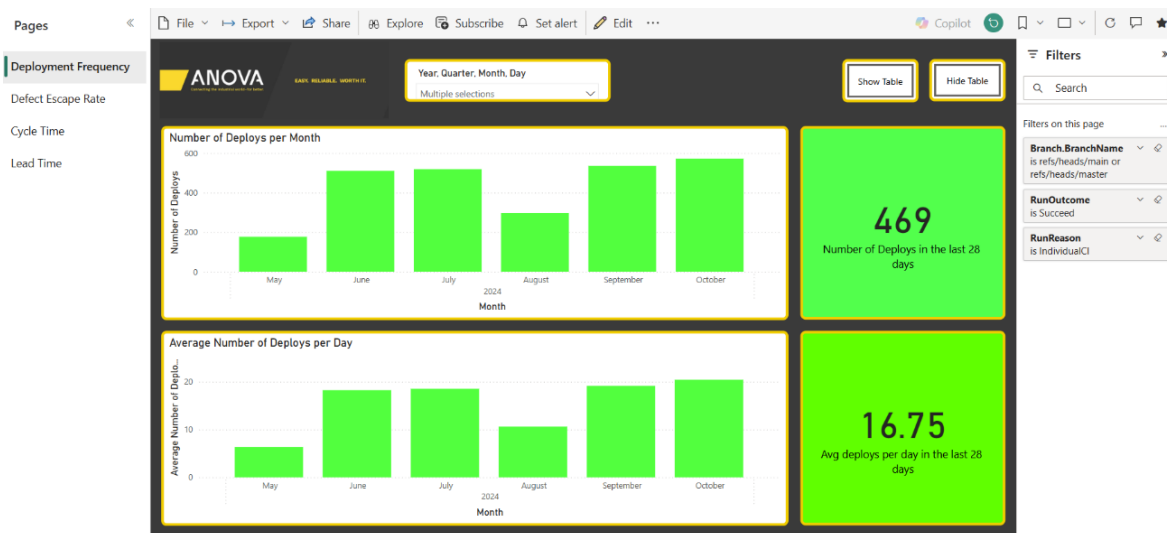


Figura 4.21. Dashboard da Frequência de Implementações (Deployment Frequency), ajustada após avaliação

DevOps and (Software) Operational Excellence Metrics

- *Dashboard da Taxa de Defeitos em Produção (DASH 2)*

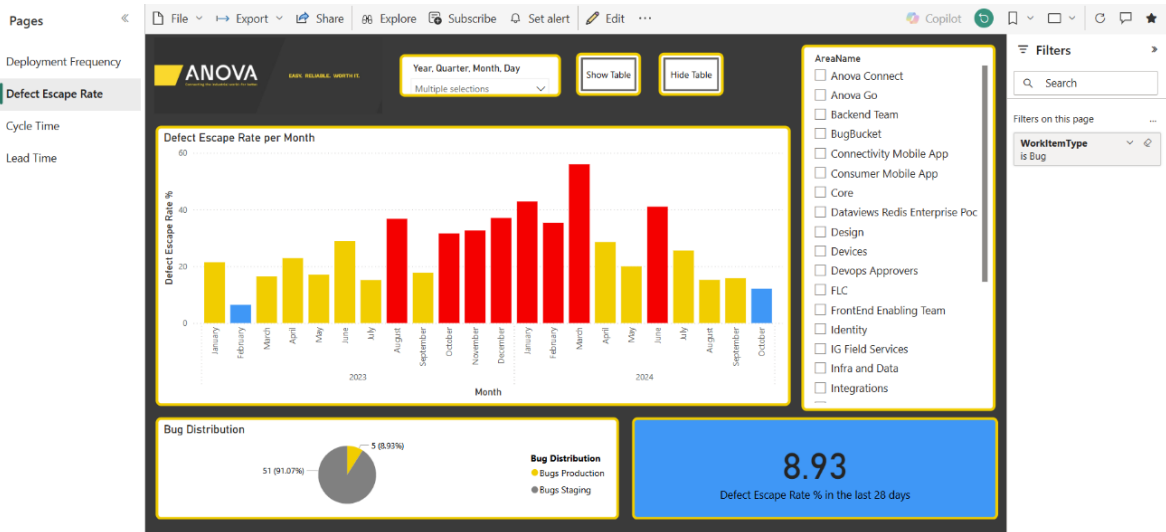


Figura 4.22. Dashboard da Taxa de Defeitos em Produção (Defect Escape Rate)

- *Dashboard do Tempo de Ciclo (DASH 3)*

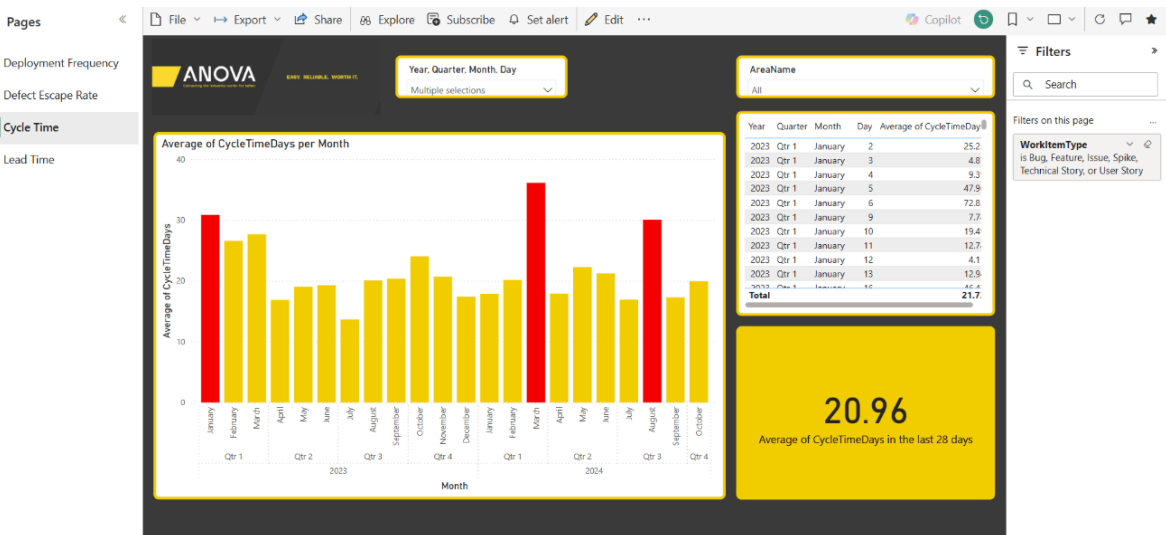


Figura 4.23. Dashboard do Tempo de Ciclo (Cycle Time)

DevOps and (Software) Operational Excellence Metrics

- *Dashboard do Tempo de Espera (DASH 4)*

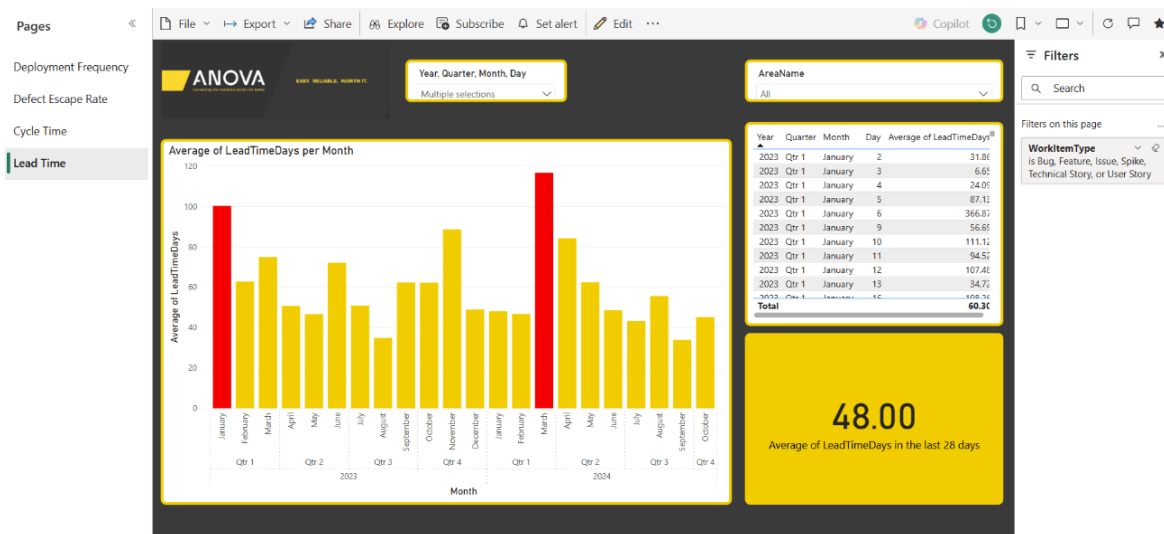


Figura 4.24. Dashboard do Tempo de Espera (Lead Time)

4.5.2 Definição dos elementos visuais

Para definir a cor dos elementos visuais e a respetiva alteração consoante os valores apresentados, recorreu-se a formatação condicional. Foram estabelecidas regras específicas para cada indicador, de modo a adaptar a representação gráfica aos dados. Apesar de estas regras terem sido aplicadas a todos os cartões e gráficos de colunas agrupadas nos *dashboards*, apenas alguns exemplos serão ilustrados nas figuras seguintes.

DevOps and (Software) Operational Excellence Metrics

Color - Effects - Background ✕

Format style
Rules

What field should we base this on?
Number of Deploys in the last 28 days

Rules ↕ Reverse color order + New rule

If value	>=	0	Number	and	<	10	Number	then	Red	↑ ↓ ✕
If value	>=	10	Number	and	<=	30	Number	then	Yellow	↑ ↓ ✕
If value	>	30	Number	and	<=	100	Number	then	Blue	↑ ↓ ✕
If value	>	100	Number	and	<=	1e+98	Number	then	Green	↑ ↓ ✕

[Learn more about conditional formatting](#) OK Cancel

Figura 4.25. Exemplo de um código para a condição das cores consoante o valor

Color - Effects - Background ✕

Format style
Rules

What field should we base this on?
Average of CycleTimeDays in the last 28

Summarization
Average

Rules ↕ Reverse color order + New rule

If value	>=	0	Number	and	<	1	Number	then	Green	↑ ↓ ✕
If value	>=	1	Number	and	<=	7	Number	then	Blue	↑ ↓ ✕
If value	>	7	Number	and	<=	30	Number	then	Yellow	↑ ↓ ✕
If value	>	30	Number	and	<=	1e+98	Number	then	Red	↑ ↓ ✕

[Learn more about conditional formatting](#) OK Cancel

Figura 4.26. Exemplo de um código para a condição das cores consoante o valor

Color - Effects - Background ×

Format style
Rules

What field should we base this on?
Average of LeadTimeDays in the last 28

Summarization
Average

Rules ↕ Reverse color order + New rule

If value	>=	0	Number	and	<	7	Number	then	Green	↑ ↓ ×
If value	>=	7	Number	and	<=	30	Number	then	Blue	↑ ↓ ×
If value	>	30	Number	and	<=	90	Number	then	Yellow	↑ ↓ ×
If value	>	90	Number	and	<=	1e+98	Number	then	Red	↑ ↓ ×

[Learn more about conditional formatting](#) OK Cancel

Figura 4.27. Exemplo de um código para a condição das cores consoante o valor

A fase de Avaliação revelou-se essencial para garantir que a solução desenvolvida respondesse às necessidades específicas da ANOVA. As sucessivas reuniões com o *Chief Software Architect* permitiram identificar áreas de melhoria e novas funcionalidades a implementar, acrescentando valor ao sistema de BI desenvolvido. As alterações introduzidas resultaram numa ferramenta mais intuitiva e eficaz para os utilizadores. A adaptação às necessidades específicas da entidade demonstrou a flexibilidade e a capacidade de personalização do *Power BI* em diferentes contextos empresariais. Assim, o processo de avaliação não só garantiu que os *dashboards* cumpriam os requisitos iniciais, como também proporcionou uma oportunidade para refinar e enriquecer a solução, com base no feedback prático dos utilizadores.

4.6 Comunicação

Na comunicação do sistema de BI desenvolvido, a apresentação dos principais resultados do projeto é crucial, tanto do ponto de vista científico como empresarial. É nesta fase que se destaca a importância do trabalho realizado, demonstrando a utilidade, o rigor e a inovação que o projeto proporciona.

DevOps and (Software) Operational Excellence Metrics

Primeiramente, apresentaram-se os resultados obtidos durante reuniões com o *Chief Software Architect* da ANOVA. Nessas reuniões, foram detalhadas tanto as métricas *DevOps* como as análises avançadas possibilitadas pelos *dashboards* desenvolvidos, o que permitiu uma compreensão clara e prática dos dados. Além de fornecerem em tempo real os indicadores necessários para medir o progresso em direção às metas – especialmente no contexto de desenvolvimento e entrega de *software* –, estas ferramentas permitiram que as equipas ajustassem processos, alocassem recursos de forma eficiente e priorizassem iniciativas que contribuíssem diretamente para os objetivos operacionais.

A comunicação empresarial focou-se em demonstrar como a solução desenvolvida podia ser utilizada para melhorar a tomada de decisão, bem como monitorizar e alcançar os objetivos operacionais no contexto de desenvolvimento e operações de *software*. Esta abordagem prática facilitou a aceitação e adoção do sistema por parte dos *stakeholders* e dos utilizadores finais. Foi também essencial destacar os benefícios alcançados com a implementação do projeto, tais como oferecer uma visão holística e em tempo real do desempenho das equipas e dos processos, permitindo uma tomada de decisão mais informada e ágil. Estes resultados demonstraram claramente o sucesso do desenvolvimento do artefacto e a sua relevância para a área de *DevOps*.

CONCLUSÃO

A realização deste estágio curricular permitiu aplicar os conhecimentos adquiridos ao longo do percurso académico em contexto real, num projeto com impacto direto na transformação digital da organização ANOVA. O principal desafio enfrentado consistiu em conceber e implementar um *dashboard* de métricas *DevOps* que respondesse às necessidades internas de monitorização, análise e melhoria dos processos de desenvolvimento de *software*.

Ao longo do projeto, foi possível identificar e estruturar um conjunto de métricas relevantes, automatizar processos de recolha e transformação de dados, e construir visualizações interativas que facilitam a leitura e interpretação dos indicadores por diferentes *stakeholders*. A ferramenta desenvolvida com *Power BI* contribuiu significativamente para aumentar a transparência organizacional e o alinhamento entre as equipas de desenvolvimento e operações, elementos fundamentais na cultura *DevOps*. As maiores dificuldades sentidas durante a execução do projeto foram de natureza conceptual, particularmente na escolha e definição das métricas *DevOps* a apresentar.

A abordagem metodológica DSR revelou-se particularmente adequada, permitindo seguir um processo iterativo de identificação do problema, desenvolvimento da solução, demonstração e avaliação do artefacto. Esta abordagem garantiu não apenas a relevância prática da solução, mas também o rigor científico necessário à validação do projeto enquanto investigação aplicada.

Do ponto de vista profissional, este projeto representou uma oportunidade de desenvolver competências técnicas em modelação de dados, *DAX*, *ETL*, *Power BI*, bem como competências transversais como análise crítica, resolução de problemas, comunicação e trabalho colaborativo. Em termos científicos, o projeto contribuiu para o aprofundamento do conhecimento na aplicação de práticas *DevOps* em contextos empresariais reais, nomeadamente no que respeita à medição e visualização de desempenho.

Em suma, a implementação do *dashboard* de métricas *DevOps* demonstrou ser uma solução eficaz e escalável, com potencial para evoluir futuramente, nomeadamente com a introdução de funcionalidades preditivas, integração com ferramentas externas ou expansão do número de indicadores analisados. Este projeto provou que, através da tecnologia e de metodologias

DevOps and (Software) Operational Excellence Metrics

adequadas, é possível criar soluções que agregam valor à organização, promovem a melhoria contínua e preparam os profissionais para os desafios de um mercado em constante transformação.

REFERÊNCIAS

- Amaro, R., Pereira, R., & Mira da Silva, M. (2024). DevOps Metrics and KPIs: A Multivocal Literature Review. *ACM Comput. Surv.*, 56(9), 231:1-231:41. <https://doi.org/10.1145/3652508>
- Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.
- Bhardwaj, A. K., Hs, S., Pareek, P. K., Bhujang, R. K., Khan, Z., & Srinivas, C. (2023). Designing a ARRHHO with MCFSA-CNN based Models for DevOps practices in Software Organizations. *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, 1–6. <https://doi.org/10.1109/ICIICS59993.2023.10420955>
- Callanan, M., & Spillane, A. (2016). DevOps: Making It Easy to Do the Right Thing. *IEEE Software*, 33(3), 53–39.
- Chen, L. (2015). Continuous Delivery: Huge Benefits, but Challenges Too. *IEEE Software*, 32(2), 50–54. <https://doi.org/10.1109/MS.2015.27>
- DORA (DevOps Research & Assessment Team). (2022). *2022 Accelerate State of DevOps Report*. <https://dora.dev/research/2022/dora-report/>
- DORA (DevOps Research & Assessment Team). (2024). *2024 Accelerate State of DevOps Report*.
- Erich, F., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, 29(6), e1885.
- Fitzgerald, B., & Stol, K.-J. (2017). Continuous software engineering: A roadmap and agenda. *Journal of Systems and Software*, 123, 176–189.
- Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The science of lean software and DevOps: Building and scaling high performing technology organizations*. IT Revolution Press.

- Fu, M., Pasuksmit, J., & Tantithamthavorn, C. (2025). AI for DevSecOps: A Landscape and Future Opportunities. *ACM Trans. Softw. Eng. Methodol.*, 34(4), 115:1-115:61. <https://doi.org/10.1145/3712190>
- Geerts, G. (2011). A design science research methodology and its application to accounting information systems research. *International Journal of Accounting Information Systems*, 12(2), 142–151.
- Gruver, G., Young, M., & Fulghum, P. (2012). *A practical approach to large-scale agile development: How HP transformed LaserJet FutureSmart firmware*. Pearson Education.
- Gunja, S. (2021). 9 key DevOps metrics for success. *Dynatrace*. <https://www.dynatrace.com/news/blog/devops-metrics-for-success/>
- Haindl, P., & Plösch, R. (2019). *Towards Continuous Quality: Measuring and Evaluating Feature-Dependent Non-Functional Requirements in DevOps*. 91–94. <https://ieeexplore.ieee.org/document/8712354>
- Hamunen, J. (2016). *Challenges in Adopting a Devops Approach to Software Development and Operations* [Aalto University]. <http://urn.fi/URN:NBN:fi:aalto-201609083476>
- Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Humble, J., & Molesky, J. (2011). Why Enterprises Must Adopt Devops to Enable Continuous Delivery. *Cutter IT Journal*, 24(8), 6–12.
- John, W., Marchetto, G., Németh, F., Sköldström, P., Steinert, R., Meirosu, C., Papafili, I., & Pentikousis, K. (2017). Service Provider DevOps. *EEE Communications Magazine*, 55(1), 204–211.
- Katal, A., Bajoria, V., & Dahiya, S. (2019). *DevOps: Bridging the gap between development and operations*. 1–7. <https://ieeexplore.ieee.org/document/8819631>

DevOps and (Software) Operational Excellence Metrics

- Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations*. IT Revolution Press.
- Krafcik, J. (1988). Triumph of the Lean Production System. *Sloan Management Review*, 30(1), 41–52.
- Kurkela, M. (2020). *DevOps Capability Assessment in a Software Development Team* [Kajaani University of Applied Sciences]. <http://www.theseus.fi/handle/10024/334710>
- Levita, C. (2017). *Proposta de modelo para avaliação da maturidade DevOps: Estudo de caso em empresas de grande porte* [Pontifícia Universidade Católica de São Paulo]. <https://tede2.pucsp.br/handle/handle/20164>
- Li, X., Yu, G., Chen, P., Chen, H., & Chen, Z. (2022). Going through the Life Cycle of Faults in Clouds: Guidelines on Fault Handling. *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*, 121–132. <https://doi.org/10.1109/ISSRE55969.2022.00022>
- Lwakatare, L., Kuvaja, P., & Oivo, M. (2016). *An exploratory study of devops extending the dimensions of devops with practices*. 91–99.
- March, S., & Storey, V. (2008). Design science in the information systems discipline: An introduction to the special issue on design science research. *MIS Quarterly*, 32(4), 725–730.
- Maroukian, K., & Gulliver, S. (2020). *Leading DevOps Practice and Principle Adoption*. 41–56. <https://doi.org/10.5121/csit.2020.100504>
- Maslyuk, D. (2022). *Exam Ref PL-300 Power BI Data Analyst*. Pearson Education.
- Peppers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77.

- Pequeno, J. (2022). *Estudo e aplicação de processos DevOps numa empresa de TI* [Instituto Superior de Engenharia de Coimbra]. <http://hdl.handle.net/10400.26/41694>
- Perera, P., Bandara, M., & Perer, I. (2016). *Evaluating the impact of DevOps practice in Sri Lankan software development organizations*. 281–287. <https://ieeexplore.ieee.org/document/7829932>
- Roche, J. (2019). Adopting DevOps Practices in Quality Assurance. *Communications of the ACM*, 56(11), 38–43.
- Romme, G. (2003). Making a difference: Organization as design. *Organization Science*, 14(5), 558–573.
- Russo, M., & Ferrari, A. (2019). *The Definitive Guide to DAX: Business Intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel* (2nd ed.). Microsoft Press.
- Scrum Portugal. (2016). Agile Manifesto. *Scrum Portugal*. <https://www.scrumportugal.pt/agile-manifesto/>
- Shah, J., & Dubaria, D. (2019). *NetDevOps: A New Era Towards Networking & DevOps*. 0775–0779. <https://bibliotheek.ehb.be:2055/document/8992969>
- Singh, G. (2024). 15 DevOps Metrics | To Boost KPI's and Its Adoptions. *XenonStack*. <https://www.xenonstack.com/blog/devops-metrics>
- Sousa, L. (2019). *DevOps – estudo de caso* [Instituto Superior de Contabilidade e Administração de Coimbra]. <http://hdl.handle.net/10400.26/31932>
- Sousa, L., Trigo, A., & Varajão, J. (2019, janeiro). *DevOps – foundations and perspectives*. <https://repositorium.sdum.uminho.pt/handle/1822/71299>
- Torble, T. (2019). Change Management and DevOps: Back to the Future. *ITNOW*, 61(3), 60–61.
- Trigo, A., Varajão, J., Soto-Acosta, P., Barroso, J., Molina-Castillo, F., & Gonzalvez-Gallego, N. (2010). IT Professionals: An Iberian Snapshot. *International Journal of Human Capital and Information Technology Professionals*, 1(1), 61–75.

- Trigo, A., Varajão, J., & Sousa, L. (2022). DevOps adoption: Insights from a large European Telco. *Cogent Engineering*, 9(1), 2083474. <https://doi.org/10.1080/23311916.2022.2083474>
- Trihinas, D., Pallis, G., Tryfonos, A., & Dikaiakos, M. (2018). DevOps as a Service: Pushing the Boundaries of Microservice Adoption. *IEEE Internet Computing*, 22(3), 65–71.
- Virmani, M. (2015). *Understanding DevOps & bridging the gap from continuous integration to continuous delivery*. 78–82. <https://doi.org/10.1109/INTECH.2015.7173368>
- Walls, M. (2013). *Building a DevOps culture*. O'Reilly Media.
- Wiedemann, A., Forsgren, N., Wiesche, M., Gewalt, H., & Krömer, H. (2019). Research for practice: The DevOps phenomenon. *Communications of the ACM*, 62(8), 44–49.
- Ziaei, S. (2022). *DevOps metrics, objectives and key results* [Auckland University of Technology]. <https://hdl.handle.net/10292/15688>

APÊNDICES

APÊNDICE 1. 24 PRÁTICAS DEVOPS

As 24 práticas são as seguintes (Forsgren et al., 2018):

Continuous Delivery

1. Utilizar controlo de versão (*version control*) para todos os artefactos de produção. O controlo de versões é a utilização de um sistema de controlo de versões, como o *GitHub* ou o *Subversion*, para todos os artefactos de produção, incluindo o código da aplicação, as configurações da aplicação, as configurações do sistema e scripts para automatizar a construção e configuração do ambiente.
2. Automatizar o processo de implementação (*deployment*). A automatização de *deployments* é o processo em que os *deployments* são totalmente automatizados e não requerem intervenção manual.
3. Implementar integração contínua (*continuous integration*). A integração contínua (CI) é o primeiro passo em direção à entrega contínua (CD). Esta é uma prática de desenvolvimento onde o código é regularmente incorporado e cada incorporação desencadeia um conjunto de testes rápidos para descobrir regressões e erros graves, que os *developers* corrigem imediatamente. O processo de CI cria *builds* e *packages* que no final serão implementados (*deployed*) e lançados (*released*).
4. Utilizar métodos de desenvolvimento baseados em troncos (*trunk-based development*). O desenvolvimento baseado em troncos (*trunk-based development*) tem-se mostrado como um indicador de alto desempenho no desenvolvimento e entrega de *software*. Caracteriza-se por ter menos de três ramos (*branches*) ativos num repositório de código; ramos e bifurcações com vidas muito curtas (por exemplo, menos de um dia) antes de serem integrados no ramo principal; e as equipas raramente ou nunca têm períodos de bloqueio de código (*code lock*) em que ninguém pode incorporar código (*check in code*) ou fazer pedidos de extração (*pull requests*) devido a conflitos de integração (*merging conflicts*), congelamento de código (*code freeze*) ou fases de estabilização.

5. Implementar automatização de testes (*test automation*). A automatização de testes é uma prática em que os testes de software são executados automaticamente (não manualmente) de forma contínua ao longo do processo de desenvolvimento. Conjuntos de testes eficazes são fiáveis, porque identificam falhas reais e apenas aprovam código que pode ser lançado. Note-se que os *developers* devem ser os principais responsáveis pela criação e manutenção dos conjuntos de testes (*test suites*) automatizados.
6. Apoiar a gestão de dados de teste (*test data management*). Os dados de testes (*test data*) requerem uma manutenção cuidadosa, e a gestão de dados de testes está a tornar-se uma parte cada vez mais importante dos testes automatizados. Práticas eficazes incluem ter dados adequados para executar o conjunto de testes, a capacidade de adquirir dados necessários quando exigido e a capacidade de incorporar os dados de teste no pipeline sem que os dados limitem a quantidade de testes que pode executar. No entanto, as equipas devem minimizar, sempre que possível, a quantidade de dados de teste necessários para executar testes automatizados.
7. *Shift left* na segurança. A integração da segurança nas fases de design e teste do processo de desenvolvimento de *software* é fundamental para impulsionar o desempenho das TI. A integração inclui a realização de análises de segurança às aplicações, o envolvimento da equipa de *infosec* no processo de design e demonstração das aplicações, a utilização de bibliotecas e pacotes de segurança previamente aprovados, e a realização de testes às funcionalidades de segurança como parte do conjunto de testes automatizados.
8. Implementar entrega contínua (*continuous delivery*). Entrega contínua (CD) é uma prática de desenvolvimento onde o software está num estado passível de ser implementado durante todo o seu ciclo de vida, e a equipa dá prioridade a manter o *software* num estado *deployable* em detrimento de trabalhar em novas funcionalidades. Um *feedback* rápido sobre a qualidade e *deployability* do sistema está disponível para todos os membros da equipa, e quando recebem relatórios de

que o sistema não está num estado *deployable*, são feitas correções rapidamente. Por fim, permite que o sistema seja *deployed* para a produção ou utilizador final (*end user*) a qualquer momento, sob pedido.

Arquitetura

9. Utilizar uma arquitetura *loosely coupled*. Ter uma arquitetura *loosely couple* permite que as equipas trabalhem de forma independente, sem depender de outras equipas para apoio e serviços, o que por sua vez lhes permite trabalhar rapidamente e proporcionar valor à organização.
10. Equipas com maior poder de decisão. Equipas que podem escolher quais ferramentas utilizar têm um desempenho superior na entrega contínua e conseqüentemente um melhor desempenho no desenvolvimento e entrega de *software*.

Produto e Processo

11. Recolher e implementar o *feedback* dos clientes. Para um melhor desempenho na entrega de *software*, é importante que as organizações procurem ativamente e regularmente o *feedback* dos clientes e o incorporem no *design* dos seus produtos.
12. Tornar visível o fluxo de trabalho ao longo do fluxo de valor. As equipas devem ter uma boa compreensão e visibilidade do fluxo de trabalho, desde o negócio até aos clientes, incluindo o estado dos produtos e funcionalidades.
13. Trabalhar em pequenas parcelas. As equipas devem dividir o trabalho em pequenas parcelas que possam ser concluídas numa semana ou menos. A chave está em decompor o trabalho em pequenas funcionalidades que permitam um desenvolvimento rápido, em vez de desenvolver funcionalidades complexas em ramos e lançá-las com pouca frequência. Esta ideia pode ser aplicada ao nível da funcionalidade e do produto. Um *Minimum Viable Product (MVP)* é um protótipo de um produto com as funcionalidades essenciais necessárias para ser lançado no mercado e testado pelos utilizadores. Trabalhar em pequenas parcelas permite tempos de execução curtos e ciclos de *feedback* mais rápidos.

DevOps and (Software) Operational Excellence Metrics

14. Promover e permitir às equipas inovarem e experimentarem novas ideias. A inovação é a capacidade da equipa de experimentar novas ideias, criar e atualizar funcionalidades durante o processo de desenvolvimento, sem necessidade de aprovação externa à equipa, o que lhes permite inovar rapidamente e criar valor. É particularmente impactante quando combinado com o trabalho em pequenas parcelas, a incorporação do *feedback* dos clientes e de um fluxo de trabalho visível.

Gestão *Lean* e Monitorização

15. Ter processos leves de aprovação de mudanças. Um processo de aprovação de alterações simplificado, baseado na revisão por pares (programação em par ou revisão de código dentro da equipa), proporciona um melhor desempenho de TI do que a utilização de *change approval boards* (CABs).
16. Monitorizar as aplicações e infraestruturas para melhorar a tomada de decisão. Utilizar dados provenientes de ferramentas de monitorização de aplicações e infraestruturas para melhorar a tomada de decisão.
17. Verificar proactivamente a integridade do sistema. Monitorize a integridade do sistema, utilizando *threshold* e *rate-of-change warnings*, para permitir às equipas detetar e mitigar problemas preventivamente.
18. Melhorar os processos e gerir o trabalho com a utilização de limites de *work-in-process* (WIP). A utilização de limites de WIP para gerir o fluxo de trabalho é amplamente reconhecida na comunidade *Lean*. Quando utilizados de forma eficaz, melhoram os processos, aumentam o *throughput* e tornam as restrições/limitações visíveis no sistema.
19. Visualizar o trabalho para monitorizar a qualidade e melhorar a comunicação com a equipa. Apresentações visuais, como *dashboards*, são usadas para monitorizar a qualidade e o trabalho em curso, contribuindo para um melhor desempenho na entrega de *software*.

Cultural

DevOps and (Software) Operational Excellence Metrics

20. Apoiar uma cultura generativa como a descrita por Westrum. Baseia-se numa tipologia desenvolvida por Ron Westrum, um sociólogo que estudou sistemas complexos críticos para a segurança nos domínios da aviação e cuidados de saúde. Esta medida de cultura é permite melhorias no desempenho da tecnologia da informação, desempenho organizacional e permite reduzir o *burnout*. As características distintivas desta medida incluem um bom fluxo de informação, elevada cooperação e confiança, pontes entre equipas e inquéritos conscientes.
21. Incentivar e apoiar a aprendizagem. Criar um ambiente e uma cultura onde os membros da equipa são incentivados a adquirir novos conhecimentos, a aprimorar as suas habilidades e a manterem-se atualizados com as melhores práticas do setor.
22. Apoiar e facilitar a colaboração entre equipas. Promover e criar um ambiente onde diferentes equipas de possam trabalhar juntas de forma eficiente e coordenada, para abordar desafios complexos e entregar *software* de alta qualidade.
23. Disponibilizar recursos e ferramentas que tornem o trabalho significativo. Ter acesso às ferramentas e recursos necessários para desempenhar o trabalho de forma eficaz, melhora a satisfação e a performance da equipa.
24. Apoiar ou incorporar liderança transformacional. A liderança transformacional apoia e fortalece o trabalho técnico e processual, que desempenham um papel essencial no *DevOps*. É composta por cinco fatores: visão, estimulação intelectual, comunicação inspiradora, liderança de apoio e reconhecimento pessoal.