Repositório Institucional UNESP

RESSALVA

Atendendo solicitação da autora, o texto completo deste documento será disponibilizado somente a partir de 28/08/2028.

unesp

UNIVERSIDADE ESTADUAL PAULISTA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



PROMOBD: UM PROCESSO DE MODERNIZAÇÃO DE SISTEMAS LEGADOS PARA ARQUITETURA DE MICROSSERVIÇOS BASEADO NA NOTAÇÃO BPMN E NA ABORDAGEM DOMAIN-DRIVEN DESIGN

NATHALIA RODRIGUES DE ALMEIDA

INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS EXATAS

RIO CLARO

UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Nathalia Rodrigues de Almeida

ProMoBD: Um Processo de Modernização de Sistemas Legados para Arquitetura de Microsserviços baseado na Notação BPMN e na Abordagem *Domain-Driven Design*

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Frank José Affonso

Rio Claro – SP
2026

A447p

Almeida, Nathalia Rodrigues de

ProMoBD: um processo de modernização de sistemas legados para arquitetura de microsserviços baseado na notação BPMN e na abordagem Domain-Driven Design / Nathalia Rodrigues de Almeida. -- Rio Claro, 2026

184 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro

Orientador: Frank José Affonso

1. Ciência da computação. 2. Engenharia de software. 3. Arquitetura orientada a serviços (Computação). 4. Modelagem de processos. 5. Projeto de sistemas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

IMPACTO POTENCIAL DESTA PESQUISA

Esta pesquisa propõe o Processo ProMoBD, voltado a apoiar a modernização de sistemas legados para a arquitetura de microsserviços com base na notação BPMN e na abordagem DDD. No contexto da Ciência e Engenharia de Software, o trabalho contribui para preencher uma lacuna identificada na literatura, relacionada à fragmentação das iniciativas de modernização e à ausência de processos capazes de contemplar, de maneira sistemática e integrada, as diferentes atividades envolvidas nessa transição arquitetural. Como impacto econômico potencial, sua aplicação pode contribuir para a redução dos riscos, dos custos de manutenção e dos esforços associados à evolução de sistemas legados, além de favorecer maior escalabilidade, disponibilidade e flexibilidade tecnológica. No âmbito educacional, o ProMoBD pode ser utilizado como referência para a formação e a capacitação de pesquisadores e profissionais envolvidos em iniciativas de modernização de software. Por fim, a integração entre BPMN, DDD e arquitetura de microsserviços favorece a inovação e a disseminação de práticas alinhadas às tendências contemporâneas da Engenharia de Software, contribuindo para a sustentabilidade tecnológica e para a evolução contínua dos sistemas organizacionais.

POTENTIAL IMPACT OF THIS RESEARCH

This research proposes the ProMoBD Process, which supports the modernization of legacy systems to a microservices architecture using BPMN notation and the DDD approach. In the context of Computer Science and Software Engineering, this work addresses a gap identified in the literature, namely the fragmentation of modernization initiatives and the lack of processes that systematically and integratively encompass the activities involved in this architectural transition. As a potential economic impact, its application may reduce risks, maintenance costs, and the effort associated with evolving legacy systems, while also promoting greater scalability, availability, and technological flexibility. In the educational context, ProMoBD can serve as a reference for the education and training of researchers and professionals involved in software modernization initiatives. Finally, the integration of BPMN, DDD, and microservices architecture promotes innovation and the dissemination of practices aligned with contemporary trends in Software Engineering, contributing to technological sustainability and the continuous evolution of organizational systems.

UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Nathalia Rodrigues de Almeida

ProMoBD: Um Processo de Modernização de Sistemas Legados para Arquitetura de Microsserviços baseado na Notação BPMN e na Abordagem *Domain-Driven Design*

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Comissão Examinadora

- Prof. Dr. Frank José Affonso (Orientador)
  Departamento de Estatística, Matemática Aplicada e Computação
  Universidade Estadual Paulista "Júlio De Mesquita Filho" (UNESP) - Câmpus de Rio Claro
- Profa. Dra. Cristiane Aparecida Lana
  Departamento de Tecnologia
  Universidade Federal de Lavras – Câmpus de São Sebastião do Paraíso
- Prof. Dr. Daniel Carlos Guimarães Pedronette
  Departamento de Estatística, Matemática Aplicada e Computação
  Universidade Estadual Paulista "Júlio De Mesquita Filho" (UNESP) - Câmpus de Rio Claro

Conceito: Aprovado

Rio Claro (SP), 28 de agosto de 2026

Aos meus pais, que me ensinaram a ser forte.
A mim mesma, prova de que aprendi.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, pela oportunidade e pela capacidade de aprender algo novo a cada dia, pela saúde e por todas as bênçãos que me permitiram chegar até aqui.

Aos meus pais, Fabio Rodrigues de Almeida e Kellen Cristina Luciano de Almeida, dedico esta dissertação e a minha mais profunda gratidão. Com muito esforço, amor e dedicação, fizeram tudo o que estava ao alcance deles para que eu pudesse estudar e seguir os caminhos que escolhi.

À minha irmã, Carolina Rodrigues de Almeida, que, além de irmã, é também minha melhor amiga, agradeço por estar ao meu lado em todos os momentos, sempre pronta a me estender a mão. Sua presença, seu cuidado e sua companhia fizeram diferença em muitos momentos dessa caminhada. A ela, todo o meu amor e admiração.

Ao meu orientador, Prof. Dr. Frank José Affonso, deixo minha sincera e profunda gratidão. Durante todos esses anos, sua orientação foi fundamental para a minha formação. Com maestria, ensinou-me não apenas a ser uma pesquisadora melhor, contribuindo para o meu desenvolvimento técnico e acadêmico, mas também me trouxe ensinamentos que levarei para a vida. Por tudo isso, serei sempre grata. Mais do que me orientar em uma pesquisa, o professor Frank me ensinou, por meio de seu próprio exemplo, como um docente deve ser.

Por fim, agradeço aos meus amigos, que tornaram esta caminhada mais leve. Agradeço pelas conversas, pelas risadas, pelos momentos de distração quando eles se fizeram necessários e, sobretudo, por me lembrarem de que as pausas também são parte do caminho.

Já que é preciso aceitar a vida, que seja então corajosamente.

Lygia Fagundes Telles

RESUMO

Nas últimas décadas, diversos sistemas de software foram desenvolvidos para atender às necessidades operacionais das organizações. Dentre esses sistemas, uma parcela significativa foi construída com base em modelos arquiteturais que se mostram obsoletos em relação ao cenário atual da computação. Embora tais modelos apresentem vantagens, sua evolução pode resultar em limitações relacionadas à manutenção, à escalabilidade, à disponibilidade e à velocidade de entrega de novas funcionalidades. Em outra perspectiva, a arquitetura de microsserviços tem sido considerada uma alternativa viável para a modernização de software, propondo a decomposição dos sistemas legados em serviços menores, independentes e organizados em torno de capacidades específicas de negócio. Apesar da crescente adoção desse estilo arquitetural, observa-se a ausência de orientações sistematizadas e consolidadas sobre como conduzir a modernização de sistemas legados para a arquitetura de microsserviços. Essa lacuna pode resultar em processos de migração inadequados, nos quais as limitações do sistema original não são efetivamente solucionadas e novos problemas são introduzidos (como acoplamento indevido e fragmentação excessiva de funcionalidades, entre outros). Diante desse cenário, uma investigação foi conduzida com o objetivo de identificar as principais dificuldades relacionadas ao processo de modernização para a arquitetura de microsserviços. Com base nas evidências reunidas nessa investigação, foi elaborado um processo de modernização chamado ProMoBD (do inglês, *Process of Legacy System Modernization to Microservice Architecture based on BPMN and DDD*). Esse processo visa apoiar a modernização de sistemas legados para a arquitetura de microsserviços com base nos conceitos da notação BPMN (do inglês, *Business Process Model and Notation*) e da abordagem DDD (do inglês, *Domain-Driven Design*). Em resumo, esse processo visa auxiliar os envolvidos na modernização, desde a compreensão até o desenvolvimento do sistema modernizado. Diante do exposto, pode-se afirmar que esta dissertação apresenta como principal contribuição o Processo ProMoBD, concebido para apoiar a modernização de sistemas legados para a arquitetura de microsserviços. Esse processo fornece uma abordagem sistemática para a modernização, contribuindo para a padronização das atividades, a redução de riscos e o apoio à tomada de decisão ao longo do ciclo de modernização. Além do ProMoBD, acredita-se que a investigação conduzida também representa uma contribuição relevante desta dissertação, pois oferece aos interessados uma visão geral sobre desse tema de pesquisa. Por fim, a fim de avaliar o potencial do referido processo, um estudo de caso foi conduzido utilizando o sistema legado *Food To GO* (FTGO). Esse estudo de caso permitiu demonstrar a aplicabilidade do processo e a execução de suas principais etapas em um cenário representativo de modernização, sem pretender validá-lo de forma ampla.

Palavras-chave: Modernização, Sistema Legado, Arquitetura de Microsserviços, DDD, BPMN.

ABSTRACT

Over the past few decades, many software systems have been developed to meet organizations' operational needs. Among these systems, a significant portion was built on architectural models that are now obsolete in the current computing landscape. Although such models have advantages, their evolution can lead to limitations in maintenance, scalability, availability, technological flexibility, and the speed of delivering new functionalities. From another perspective, microservices architecture has been considered a viable alternative for modernizing these systems, decomposing legacy systems into smaller, independent services organized around specific business capabilities. Despite the growing adoption of this architectural style, there is a lack of systematized and consolidated guidelines for modernizing legacy systems into a microservices architecture. This gap can lead to inadequate migration processes, in which the limitations of the original system are not effectively resolved and new problems are introduced (such as greater operational complexity, improper coupling, and excessive fragmentation of functionalities, among others). Given this scenario, an investigation was conducted to identify the main difficulties related to the modernization of legacy systems to a microservices architecture. Based on the evidence gathered in this investigation, a modernization process called ProMoBD (Process of Legacy System Modernization to Microservice Architecture based on BPMN and DDD) was developed. This process aims to support the modernization of legacy systems to a microservices architecture based on the concepts of BPMN (Business Process Model and Notation) and the DDD (Domain-Driven Design) approach. In short, this process aims to assist those involved in modernization, from understanding to developing the modernized system. Thus, this dissertation's main contribution is the ProMoBD Process, designed to support the modernization of legacy systems to a microservices architecture. This process provides a systematic approach to modernization, contributing to the standardization of activities, risk reduction, and support for decision-making throughout the modernization cycle. In addition to the ProMoBD, the investigation conducted is also believed to represent a contribution of this dissertation, as it offers stakeholders an overview of this research topic. Finally, to evaluate the potential of this process, a case study was conducted using the legacy system *Food To GO* (FTGO). This case study demonstrated the applicability of the process and the execution of its main steps in a representative modernization scenario, without claiming broad validation.

Keywords: Modernization, Legacy System, Microservice Architecture, DDD, BPMN.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura monolítica 19
Figura 2 – Integração de software 20
Figura 3 – Visão geral do padrão MVC 24
Figura 4 – Arquitetura de microsserviços 26
Figura 5 – Visão geral da arquitetura baseada em DDD 30
Figura 6 – Distribuição das iniciativas por iniciativa 36
Figura 7 – Distribuição dos estudos pelos conceitos e nível de conhecimento 38
Figura 8 – Distribuição de estudos por ano de publicação 40
Figura 9 – Distribuição dos estudos pela natureza das publicações 42
Figura 10 – Distribuição dos estudos pelo nível de maturidade das publicações 43
Figura 11 – Distribuição dos estudos pelo domínio de aplicação 44
Figura 12 – Distribuição dos estudos pelas etapas relacionada à abordagem DDD 48
Figura 13 – Nuvem de palavras sobre os fatores motivadores relacionados ao uso da abordagem DDD na modernização de sistemas legados para a arquitetura de microsserviços 49
Figura 14 – Dificuldades reportadas nos estudos analisados 53
Figura 15 – Visão geral do Processo de Modernização ProMoBD 62
Figura 16 – Linguagem ubíqua e a comunicação entre os envolvidos 72
Figura 17 – Formato sugerido de uma *user story* 74
Figura 18 – Visão geral da identificação dos processos de negócio 78
Figura 19 – Modelo BPMN *as-is* 81
Figura 20 – Processo de construção de Modelos BPMN *to-be* 83
Figura 21 – Mapeamento de *bounded context* para microsserviços 87
Figura 22 – Definição dos candidatos a microsserviços 90
Figura 23 – Visão geral de um pipeline CI/CD para desenvolvimento dos microsserviços 101
Figura 24 – Processo de migração gradual do microsserviço MSPagamento com *Feature Toggle* 105
Figura 25 – Diagrama de classes simplificado do FTGO 118
Figura 26 – Modelo BPMN *as-is* do processo *Order Management* 136
Figura 27 – Modelo BPMN *to-be* do processo *Order Management* (baseado em eventos) 138

LISTA DE TABELAS

Tabela 1 – Distribuição dos estudos pelos locais de publicação 41
Tabela 2 – Síntese da Etapa 1.1 65
Tabela 3 – Síntese da Etapa M1.2 68
Tabela 4 – Síntese da Etapa 1.3 69
Tabela 5 – Síntese da Etapa 2.1 73
Tabela 6 – Síntese da Etapa 2.2 76
Tabela 7 – Síntese da Etapa 3.1 80
Tabela 8 – Síntese da Etapa 3.2 82
Tabela 9 – Síntese da Etapa 3.3 85
Tabela 10 – Exemplo de especificação do Microsserviço Pagamento 95
Tabela 11 – Síntese da Macroatividade 4 95
Tabela 12 – Síntese da Etapa 5.1 99
Tabela 13 – Síntese da Etapa 5.2 102
Tabela 14 – Síntese da Etapa 5.3 106
Tabela 15 – Síntese da Etapa 6.1 109
Tabela 16 – Contexto atual do sistema FTGO 119
Tabela 17 – Camadas arquiteturais do sistema FTGO 119
Tabela 18 – Artefatos produzidos durante a engenharia reversa do FTGO 122
Tabela 19 – Síntese da avaliação de qualidade do sistema FTGO 124
Tabela 20 – Subdomínios e *bounded contexts* do FTGO 127
Tabela 21 – Glossário da linguagem ubíqua do FTGO 128
Tabela 22 – Histórias de usuário priorizadas para o FTGO 129
Tabela 23 – Caso de uso: Realizar pedido 131
Tabela 24 – Processos de negócio identificados para o FTGO 134
Tabela 25 – Comparação entre os modelos *as-is* e *to-be* 138
Tabela 26 – Lista inicial de microsserviços candidatos do sistema FTGO 143
Tabela 27 – Especificação final dos microsserviços do FTGO 146
Tabela 28 – Ciclos de migração do FTGO 148
Tabela 29 – Lista de Bases de Pesquisa 175
Tabela 30 – *String* de Pesquisa 175
Tabela 31 – Lista de Bases de Pesquisa 177
Tabela 32 – Formulário de Extração de Dados 178
Tabela 33 – Lista Final de Estudos 180

LISTA DE ABREVIATURAS E SIGLAS

ACID	*Atomicity, Consistency, Isolation and Durability*
ADR	*Architecture Decision Record*
API	*Application Programming Interface*
BPMN	*Business Process Model and Notation*
CDC	*Change Data Capture*
CD	*Continuous Delivery*
CI	*Continuous Integration*
DDD	*Domain-Driven Design*
DLQ	*Dead Letter Queue*
DSL	*Domain-Specific Language*
ECR	*Elastic Container Registry*
ELK	*Elasticsearch, Logstash and Kibana*
FTGO	*Food To Go*
IaC	*Infrastructure as Code*
ID	*IDentifier*
IoT	*Internet of Things*
JPA	*Java Persistence API*
JWT	*JSON Web Token*
MSA	*Microservices Architecture*
MVC	*Model-View-Controller*
NPS	*Net Promoter Score*
OMG	*Object Management Group*
OWASP	*Open Web Application Security Project*
PDCA	*Plan, Do, Check and Act*
ProMoBD	*Process of Legacy System Modernization to Microservice Architecture based on BPMN and DDD*
QP	Questão de Pesquisa
REST	*Representational State Transfer*
RPC	*Remote Procedure Call*
SLI	*Service Level Indicator*
SLO	*Service Level Objective*
SMS	*Systematic Mapping Study*
TI	Tecnologia da Informação

SUMÁRIO

1 INTRODUÇÃO 11
1.1 Contextualização 11
1.2 Motivação 14
1.3 Objetivos 15
1.4 Contribuições 16
1.5 Organização da Monografia 16

2 FUNDAMENTAÇÃO TEÓRICA 18
2.1 Integração e modernização de software 19
2.2 Arquitetura de Software 22
2.3 Arquitetura de microsserviços 25
2.4 *Business Process Model and Notation* 27
2.5 *Domain-driven design* 29
2.6 Considerações finais 32

3 MODERNIZAÇÃO DE SISTEMAS LEGADOS PARA ARQUITETURA DE MICROSSERVIÇOS 34
3.1 Mapeamento sistemático 34
3.1.1 Iniciativas para modernização de sistemas legados para microsserviços . 35
3.1.2 Conceito base utilizado 37
3.1.3 Características demográficas dos estudos 39
3.1.4 Domínios de aplicação 44
3.1.5 Uso abordagem DDD na modernização de software 46
3.1.6 Fatores motivadores para adoção de DDD 48
3.1.7 Desafios enfrentados 51
3.2 Discussão de resultados 52
3.3 Considerações finais 58

4 PROCESSO DE MODERNIZAÇÃO 60
4.1 Considerações iniciais 60
4.2 Concepção do processo 60
4.3 Atividades do processo 61
4.3.1 M1. Análise do Sistema Legado 63
M1.1. Identificação do contexto atual 63

M1.2. Engenharia reversa . . . 65
M1.3. Avaliação da qualidade dos artefatos de documentação . . . 67
4.3.2 M2. Levantamento de Requisitos baseada em DDD . . . 69
M2.1. Modelagem de domínio baseada em DDD . . . 70
M2.2. Modelagem baseada em histórias de usuário . . . 73
4.3.3 M3. Modelagem de Processos de Negócio com BPMN . . . 76
M3.1. Identificação dos processos de negócio . . . 77
M3.2. Construção dos modelos BPMN as-is . . . 79
M3.3. Construção dos modelos BPMN to-be . . . 82
4.3.4 M4. Definição da Arquitetura de Microsserviços . . . 85
M4.1. Mapeamento dos bounded contexts para microsserviços . . . 86
M4.2. Estratégias de decomposição . . . 88
M4.3. Definição dos candidatos a microsserviços . . . 89
M4.4. Padrões Arquiteturais e de Integração . . . 93
4.3.5 M5. Implementação Incremental e Migração . . . 95
M5.1. Planejamento detalhado da implementação incremental . . . 96
M5.2. Preparação dos ambientes de desenvolvimento e produção . . . 99
M5.3. Implementação iterativa e incremental dos microsserviços . . . 102
4.3.6 M6. Monitoramento, Avaliação e Melhoria Contínua . . . 106
M6.1. Definição de métricas e indicadores de qualidade . . . 107
M6.2. Monitoramento contínuo do ambiente . . . 109
M6.3. Avaliação e *feedback* com *stakeholders* . . . 110
M6.4. Ciclo de melhoria contínua . . . 111
4.4 Considerações finais . . . 112

5 ESTUDO DE CASO . . . 114
5.1 Descrição do sistema legado FTGO . . . 114
5.2 Aplicando o Processo ProMoBD . . . 115
5.2.1 EC-M1. Análise do Sistema Legado . . . 116
EC-M1.1. Identificação do contexto atual . . . 116
EC-M1.2. Engenharia reversa . . . 120
EC-M1.3. Avaliação da qualidade dos artefatos de documentação . . . 121
5.2.2 EC-M2. Levantamento e Documentação dos Requisitos em DDD . . . 125
M2.1. Modelagem de domínio baseada em DDD . . . 125
M2.2. Modelagem baseada em histórias de usuário . . . 127
5.2.3 EC-M3. Modelagem dos Processos com BPMN . . . 132
M3.1. Identificação dos processos de negócio . . . 132
M3.2. Construção dos modelos BPMN *as-is* . . . 133

M3.3. Construção dos modelos BPMN *to-be* 135
5.2.4 EC-M4. Definição da Arquitetura de Microsserviços 139
M4.1. Mapeamento dos *bounded contexts* para microsserviços 140
M4.2. Estratégias de decomposição 140
M4.3. Definição dos candidatos a microsserviços 142
M4.4. Padrões arquiteturais e de integração 145
5.2.5 EC-M5. Implementação Incremental e Migração 147
M5.1. Planejamento detalhado da implementação incremental 147
M5.2. Preparação dos ambientes de desenvolvimento e produção 149
M5.3. Implementação iterativa e incremental dos microsserviços 151
5.2.6 EC-M6. Monitoramento, Avaliação e Melhoria Contínua 153
M6.1. Definição de métricas e indicadores de qualidade 154
M6.2. Monitoramento contínuo do ambiente 155
M6.3. Avaliação e *feedback* com *stakeholders* 156
M6.4. Ciclo de melhoria contínua 157
5.3 Considerações finais 159

6 CONCLUSÃO 160
6.1 Contribuições da dissertação 161
6.2 Dificuldades e limitações 162
6.3 Trabalhos futuros 163

REFERÊNCIAS 165

APÊNDICE A – PROTOCOLO DE PESQUISA 172
A.1 Introdução 172
A.2 Questões de Pesquisa 173
A.3 Estratégia de Pesquisa 174
A.4 Extração e Síntese de Dados 176
A.5 Limitações 177
A.6 Síntese dos Resultados 179

APÊNDICE B – LISTA DE ESTUDOS PRIMÁRIOS 180

DADOS CURRICULARES 185

1 INTRODUÇÃO

1.1 Contextualização

Os sistemas de software vêm desempenhando um papel preponderante na sociedade ao longo das últimas quatro décadas, podendo ser encontrados em diferentes segmentos, tais como: instituições financeiras, setor aeroportuário, comércio em geral, entre outros. Ao observar o referido período, especificamente durante as décadas de 1970 e 1980, nota-se que o processamento das informações nesses sistemas era realizado de maneira centralizada em grandes computadores (ou seja, *mainframes*) (PRESSMAN; MAXIM, 2019). No que diz respeito à estrutura e ambiente de execução, vale destacar que esses sistemas foram projetados com base nos recursos computacionais de hardware e software disponíveis à época de seu desenvolvimento, sendo mantidos ao longo dos anos ou permanecendo em operação até os dias atuais (SOMMERVILLE, 2011; PRESSMAN; MAXIM, 2019).

Conforme reportado por Lewis e Fowler (2014), as necessidades da sociedade nos últimos 20 anos tornaram esses sistemas significativamente mais complexos, tanto sob a perspectiva interna do software (por exemplo, estruturas internas, modelos arquiteturais, entre outros) quanto no que diz respeito ao ambiente de execução no qual estão inseridos (por exemplo, servidores, banco de dados, entre outros). Em um cenário entre 2000 e 2015, com o intuito de atender às demandas emergentes de mercado, tais sistemas foram integrados a ambientes mais recentes via APIs (do inglês, *Application Programming Interface*). De acordo com Pressman (2014), a integração de software foi (ou tem sido) uma alternativa viável para disponibilizar um sistema "antigo", desenvolvido com recursos computacionais obsoletos, para um cenário de execução mais recente dotado de recursos computacionais mais modernos. Para promover a integração de sistemas, os desenvolvedores devem preservar o sistema original em seu ambiente de execução nativo e construir uma camada de integração intermediária, a qual deve ser responsável por disponibilizar esse sistema para interfaces e recursos mais modernos. Dessa forma, pode-se dizer que esses sistemas podem atender às novas demandas de mercado preservando sua lógica de negócio em seu ambiente nativo ao qual foram projetados (PRESSMAN; MAXIM, 2019).

Diante do cenário apresentado e visando estabelecer um alinhamento conceitual nesta dissertação, a definição de sistemas legados estabelecida por Pressman e Maxim (2019) será adotada deste ponto em diante. De acordo com o referido autor, sistema legado pode ser definido como um sistema de software desenvolvido com recursos computacionais obsoletos (por exemplo, modelos arquiteturais, linguagens de programação, bancos de dados, entre outros) se comparado ao cenário computacional vigente. Em relação à estrutura interna, vale ressaltar

que esse conceito não se restringe apenas aos sistemas projetados com base em uma arquitetura monolítica, podendo abranger outros modelos arquiteturais mais recentes. Conforme reportado por Lewis e Fowler (2014), uma aplicação monolítica é construída como uma única unidade indivisível, a qual é composta pelos seguintes elementos: interface de usuário, lógica de negócio e camada de dados estão fortemente acopladas e implantadas conjuntamente. Além disso, pode-se dizer que esses sistemas requerem atividade de manutenção contínua para que os (novos) requisitos de negócio sejam atendidos (LEWIS; FOWLER, 2014).

Fazendo uma breve análise dos sistemas supracitados, os sistemas legados, desenvolvido sob uma arquitetura monolítica podem oferecer vantagens significativas, especialmente nas fases iniciais do desenvolvimento: a simplicidade de um único repositório de código facilita a navegação e a compreensão do sistema, os testes de ponta a ponta são mais diretos, e o processo de *deploy* é simplificado por envolver um único artefato (NEWMAN, 2019). Além disso, a consistência transacional também é um benefício notável, uma vez que um único banco de dados permite a aplicação nativa do modelo ACID (Atomicidade, Consistência, Isolamento e Durabilidade) (RICHARDSON, 2018). Apesar de suas vantagens, esses sistemas apresentam limitações significativas à medida que a complexidade organizacional é elevada. Por estarem organizados como uma única unidade fortemente acoplada, grandes sistemas tornam-se progressivamente difíceis de serem mantidos, atualizados e escalados (NEWMAN, 2019). À medida que a base de código aumenta, o tempo de *deploy* cresce consideravelmente, uma vez que qualquer alteração, mesmo que pequena, exige a reimplantação da aplicação como um todo. Além disso, o acoplamento forte também pode comprometer a resiliência do sistema, pois uma falha em um componente pode ser propagada por toda a aplicação em um efeito cascata, comprometendo sua disponibilidade. A adoção de novas tecnologias também torna-se restrita, pois toda a aplicação está vinculada a uma única *stack* tecnológica (RICHARDSON, 2018). Diante desse cenário, surge a arquitetura de microsserviços (do inglês, *Microservice Architecture* – MSA) como uma alternativa para contornar as limitações e/ou desafios impostos pelo crescimento dos sistemas legados desenvolvidos (ou não) em uma arquitetura monolítica.

Na visão de Lewis e Fowler (2014), a arquitetura de microsserviços representa uma abordagem embasada no desenvolvimento de aplicações compostas por um conjunto de pequenos serviços. Esse estilo arquitetural prevê que cada serviço seja projetado com base nas capacidades de negócio, seja executado em seu próprio processo e se comunique por meio de mecanismos leves, e seja implantado de maneira independente mediante rotinas totalmente automatizadas. Como pode ser observado, essa definição evidencia o potencial desse estilo arquitetural em lidar com as adversidades inerentes aos sistemas legados apresentadas nesta seção. Em outras palavras, ao adotar essa arquitetura, os desenvolvedores podem contornar esses gargalos por meio da construção de serviços reduzidos e independentes. Sob essa perspectiva, pode-se dizer que essa organização permite viabilizar a realização de *deploys* mais ágeis, a atuação de times

autônomos e o emprego de uma escalabilidade granular (ou seja, a capacidade de dimensionar isoladamente apenas os componentes que demandam recursos, sem a necessidade de escalar a aplicação em sua totalidade) (RICHARDSON, 2018; NEWMAN, 2019).

Em outra perspectiva, Richardson (2018) comenta que a arquitetura de microsserviços também apresenta limitações relevantes. Nesse sentido, observa-se que a comunicação entre os microsserviços necessita ser criteriosamente projetada, uma vez que, diferentemente do que ocorre no modelo monolítico, não existem garantias nativas de atomicidade para transações que envolvem múltiplos componentes. Com a descentralização do armazenamento e a adoção de bancos de dados isolados por microsserviços, compromete-se o suporte automático ao modelo transacional ACID, tornando-se imperativa a aplicação de estratégias específicas para assegurar a integridade e a consistência dos dados ao longo de todo o sistema (RICHARDSON, 2018). Além disso, soma-se a essa adversidade a evidência de que a operação contínua de um sistema distribuído passa a demandar um nível significativamente maior de maturidade técnica por parte dos times de desenvolvimento e parte da organização. Para isso, torna-se essencial a incorporação de mecanismos para monitoramento, rastreamento distribuído e orquestração de serviços, elementos que tendem a elevar a complexidade operacional do ambiente (NEWMAN, 2019).

Apesar de existirem diversos contrapontos, a modernização de sistemas legados para a arquitetura de microsserviços tem ganhado evidência significativa nos últimos cinco anos, conforme evidências reunidas no estudo de Almeida *et al.* (2024) e no Capítulo 3. Essas evidências revelaram que a modernização de software tem se consolidado como uma estratégia viável para promover a transição de sistemas legados em direção a soluções respaldadas por tecnologias contemporâneas, modelos arquiteturais atualizados e cenários computacionais vigentes (LEWIS; FOWLER, 2014). Em outras palavras, o estilo arquitetural de microsserviços destaca-se como uma alternativa capaz de propiciar o desacoplamento nos níveis estrutural, funcional e de dados, além de ser aderente às práticas de DevOps (*Development and Operations*). Em resumo, essas práticas demandam ciclos de entrega e lançamento progressivamente mais ágeis mediante o desacoplamento estrutural, funcional e de dados (ABGAZ *et al.*, 2023).

Ainda sobre o estudo conduzido por Almeida *et al.* (2024), nota-se que a abordagem de desenvolvimento de software DDD (do inglês, *Domain-Driven Design*) (EVANS, 2004; VERNON, 2016) tem sido adotada para apoiar o desenvolvimento (e/ou modernização) de software alinhada às regras de negócio. No contexto específico da arquitetura de microsserviços, a filosofia DDD tem auxiliado diretamente o processo de decomposição do sistema a partir de três conceitos fundamentais: (i) *Bounded Context*, que permite delimitar as fronteiras naturais entre os domínios do negócio, guiando o isolamento e a definição dos serviços; (ii) Linguagem Ubíqua, que promove o estabelecimento de um vocabulário único e compartilhado entre os times de desenvolvimento e os especialistas do domínio, reduzindo inconsistências e ambiguidades;

e (iii) *Context Map*, que viabiliza a visualização clara dos relacionamentos e das dependências existentes entre os diferentes contextos do sistema. Dessa forma, pode-se dizer que a abordagem DDD representa um meio viável capaz de nortear a modernização de sistemas legados (monolíticos ou não) para a arquitetura de microsserviços.

6 Conclusão

Esta dissertação apresentou o ProMoBD, um processo para apoiar a modernização de sistemas legados para a arquitetura de microsserviços baseado na notação BPMN e na abordagem de desenvolvimento de software DDD. Esse processo foi organizado de forma a oferecer uma abordagem incremental e orientada ao domínio de negócio, contemplando não apenas aspectos técnicos, mas também organizacionais, reconhecendo que a modernização bem-sucedida depende da integração entre tecnologia, pessoas e processos. Nesse sentido, vale ressaltar que cada macroatividade foi estruturada em um conjunto de etapas que orientam desde a análise do sistema legado até o monitoramento e a melhoria contínua do software modernizada. Em síntese, o ProMoBD tem como meta oferecer aos interessados na atividade de modernização de sistemas legados para a arquitetura de microsserviços uma alternativa sistematizada que contempla todos os passos de um ciclo de desenvolvimento (ALMEIDA *et al.*, 2024; FÁVERO; MÁRIO; AFFONSO, 2025). Nessa direção, as evidências reunidas no Capítulo 5 revelam que, se seguido na íntegra, o processo proposto tende a reduzir os riscos inerentes à modernização, promovendo o desacoplamento gradual dos componentes legados e incentivando boas práticas arquiteturais e de engenharia de software.

De acordo com a proposta apresentada nesta dissertação, o desenvolvimento desse tipo de processo envolve uma carga elevada de conceitos e conhecimentos de diferentes áreas. Inicialmente, no Capítulo 2 foram reportados os fundamentos teóricos necessários à compreensão do trabalho, abrangendo os conceitos de arquitetura de software, as características da arquitetura monolítica, a arquitetura orientada a serviços e uma descrição detalhada da arquitetura de microsserviços, destacando seus benefícios, desafios e melhores práticas. Adicionalmente, foram apresentados os princípios da abordagem DDD e da notação BPMN, que constituem os pilares metodológicos do processo proposto. Em seguida, no Capítulo 3 foi apresentado uma investigação da literatura sobre modernização de sistemas legados para a arquitetura de microsserviços, conduzido com base em 45 estudos primários, cujos resultados evidenciaram lacunas e oportunidades que motivaram a concepção do Processo ProMoBD. No Capítulo 4 foram reportados os detalhes do processo proposto, o qual foi organizado em seis macroatividades que abrangem desde a análise do sistema legado até o monitoramento e a melhoria contínua do sistema modernizado. No Capítulo 5 foi apresentado o estudo de caso conduzido para avaliar a aplicabilidade do processo proposto com base no sistema FTGO. Por fim, neste capítulo são apresentadas as conclusões desta dissertação de mestrado da seguinte maneira: na Seção 6.1 são descritas as contribuições desta dissertação; na Seção 6.2 são reportadas as principais dificuldades enfrentadas durante a condução deste projeto de mestrado e suas limitações; e na Seção 6.3

são apresentadas as perspectivas de trabalhos futuros.

REFERÊNCIAS

ABGAZ, Y. *et al.* Decomposition of monolith applications into microservices architectures: A systematic review. *IEEE Transactions on Software Engineering*, v. 49, n. 8, p. 4213–4242, Aug 2023. ISSN 1939-3520. Disponível em: <https://doi.org/10.1109/TSE.2023.3287297>. Citado na página 13.

ALMEIDA, N. R. *et al.* Modernization of legacy systems to microservice architecture: A tertiary study. In: INSTICC. *The 23rd International Conference on Enterprise Information Systems*. [S.l.]: SciTePress, 2024. p. 1–12. ISBN 978-989-758-509-8. ISSN 2184-4992. Citado 13 vezes na(s) página(s) 13, 14, 15, 60, 72, 74, 100, 103, 104, 107, 108, 160 e 161.

ALMEIDA, N. R. d.; FÁVERO, L. F.; AFFONSO, F. J. From domain-driven design to microservice architecture: A systematic mapping focused on modernization of legacy systems. In: *Proceedings of the 20th Iberian Conference on Information Systems and Technologies (CISTI 2025). Artigo apresentado na conferência*. Lisbon, Portugal: Artigo não publicado nos anais da conferência, 2025. Citado 2 vezes na(s) página(s) 15 e 72.

ALSOLAMI, F. Domain-driven microservice architecture for designing modern applications. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY*, v. 20, n. 6, p. 152–159, JUN 30 2020. ISSN 1738-7906. Disponível em: <http://paper.ijcsns.org/07_book/202006/20200617.pdf>. Citado 8 vezes na(s) página(s) 15, 68, 86, 89, 92, 97, 103 e 106.

ALWIS, A. A. C. D. *et al.* Business object centric microservices patterns. In: PANETTO, H. *et al.* (Ed.). *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*. Cham: Springer International Publishing, 2019. p. 476–495. ISBN 978-3-030-33246-4. Disponível em: <https://doi.org/10.1007/978-3-030-33246-4_30>. Citado 2 vezes na(s) página(s) 92 e 93.

Amazon Web Services. *Prioritization and migration strategy*. s.d. AWS Prescriptive Guidance. Disponível em: <https://docs.aws.amazon.com/prescriptive-guidance/latest/application-portfolio-assessment-guide/prioritization-and-migration-strategy.html>. Acesso em: 12 out. 2026. Citado na página 98.

AYAS, H. M.; LEITNER, P.; HEBIG, R. The migration journey towards microservices. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, v. 13126 LNCS, p. 20 – 35, 2021. ISSN 03029743. Disponível em: <https://doi.org/10.1007/978-3-030-91452-3_2>. Citado 2 vezes na(s) página(s) 64 e 66.

BAI, R. *et al.* Research on the forecast of workshop production abnormity under cloud manufacturing environment. In: *Proceedings of the ACM Turing Award Celebration Conference - China*. New York, NY, USA: Association for Computing Machinery, 2021. (ACM TURC '21), p. 224–232. ISBN 9781450385671. Disponível em: <https://doi.org/10.1145/3472634.3474076>. Citado na página 64.

BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software Architecture in Practice*. 3rd. ed. [S.l.]: Addison-Wesley Professional, 2013. ISBN 978-0321815736. Citado 3 vezes na(s) página(s) 21, 24 e 27.

BECKER, J.; KUGELER, M.; ROSEMANN, M. (Ed.). *Process Management: A Guide for the Design of Business Processes*. [S.l.]: Springer, 2011. Citado na página 28.

BEYER, B. *et al. Site Reliability Engineering: How Google Runs Production Systems*. Sebastopol, CA: O'Reilly Media, 2016. ISBN 9781491929124. Citado na página 107.

CAMILLI, M. *et al.* Actor-driven decomposition of microservices through multi-level scalability assessment. *ACM Trans. Softw. Eng. Methodol.*, Association for Computing Machinery, New York, NY, USA, v. 32, n. 5, jul 2023. ISSN 1049-331X. Disponível em: <https://doi.org/10.1145/3583563>. Citado na página 106.

CHEN, L. Microservices: Architecting for continuous delivery and devops. *IEEE Software*, v. 35, n. 2, p. 50–55, 2018. Citado 2 vezes na(s) página(s) 22 e 61.

CHIKOFSKY, E.; CROSS, J. Reverse engineering and design recovery: a taxonomy. *IEEE Software*, v. 7, n. 1, p. 13–17, 1990. Citado na página 65.

COHN, M. *User Stories Applied: For Agile Software Development*. Boston, MA: Addison-Wesley, 2004. ISBN 978-0321205681. Citado 3 vezes na(s) página(s) 74, 75 e 79.

COLANZI, T. *et al.* Are we speaking the industry language? the practice and literature of modernizing legacy systems with microservices. In: *The 15th Brazilian Symposium on Software Components, Architectures, and Reuse*. New York, NY, USA: Association for Computing Machinery, 2021. p. 61–70. ISBN 9781450384193. Disponível em: <https://doi.org/10.1145/3483899.3483904>. Citado 2 vezes na(s) página(s) 21 e 63.

CUNNINGHAM, P.; CUNNINGHAM, M. A framework for the acceptance process of district health information system for vertical health programmes. In: IIMC INTERNATIONAL INFORMATION MANAGEMENT CORPORATION. *IST-Africa 2015 Conference Proceedings*. [S.l.], 2015. Citado 2 vezes na(s) página(s) 20 e 68.

DIEPENBROCK, A.; RADEMACHER, F.; SACHWEH, S. An ontology-based approach for domain-driven design of microservice architectures. In: . [s.n.], 2017. v. 275, p. 1777 – 1791. Cited by: 9. Disponível em: <https://doi.org/10.18420/in2017_177>. Citado na página 106.

DIESTE, O.; GRIMáN, A.; JURISTO, N. Developing search strategies for detecting relevant experiments. *Empirical Software Engineering*, Kluwer Academic Publishers, Hingham, MA, USA, v. 14, n. 5, p. 513–539, 2009. ISSN 1382-3256. Citado na página 174.

DRAGONI, N. *et al.* Microservices: Yesterday, today, and tomorrow. *Springer*, p. 195–216, 2017. Citado 2 vezes na(s) página(s) 21 e 91.

DUMAS, M. *et al. Fundamentals of Business Process Management*. 2. ed. [S.l.]: Springer, 2018. ISBN 978-3662565095. Citado 7 vezes na(s) página(s) 27, 77, 80, 82, 85, 88 e 111.

EVANS, E. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. [S.l.]: Addison-Wesley Professional, 2004. ISBN 978-0321125217. Citado 20 vezes na(s) página(s) 13, 14, 18, 29, 31, 37, 38, 46, 47, 52, 58, 61, 69, 70, 71, 72, 75, 86, 87 e 90.

FÁVERO, L. F.; ALMEIDA, N. R. d.; AFFONSO, F. J. A systematic mapping study on the modernization of legacy systems to microservice architecture. *Applied System Innovation*, v. 8, n. 4, 2025. ISSN 2571-5577. Disponível em: <https://www.mdpi.com/2571-5577/8/4/86>. Citado 6 vezes na(s) página(s) 14, 15, 60, 72, 107 e 161.

FÁVERO, L. F.; MÁRIO, G. S.; AFFONSO, F. J. Micro4delphi: A process for the modernization of legacy systems in delphi to microservice architecture. In: INSTICC. *The 24rd International Conference on Enterprise Information Systems. ARTIGO ACEITO PARA PUBLICAÇÃO*. [S.l.]: SciTePress, 2025. p. 1–12. ISBN 978-989-758-509-8. ISSN 2184-4992. Citado 8 vezes na(s) página(s) 14, 15, 60, 104, 107, 108, 160 e 161.

FERNANDES, J. M.; MACHADO, R. J. Modeling and analyzing legacy systems using bpmn. In: *Proceedings of the 4th International Conference on Enterprise Information Systems (ICEIS)*. [S.l.: s.n.], 2006. Citado 2 vezes na(s) página(s) 15 e 77.

FOWLER, M. *Patterns of Enterprise Application Architecture*. [S.l.]: Addison-Wesley Professional, 2003. ISBN 978-0321127426. Citado 4 vezes na(s) página(s) 29, 32, 146 e 147.

FOWLER, M. *Refactoring: Improving the Design of Existing Code*. 2nd. ed. [S.l.]: Addison-Wesley Professional, 2018. ISBN 978-0134757599. Citado 4 vezes na(s) página(s) 18, 21, 70 e 96.

FRANCESCO, P. D.; LAGO, P.; MALAVOLTA, I. Migrating towards microservice architectures: An industrial survey. In: *IEEE International Conference on Software Architecture (ICSA)*. [s.n.], 2018. p. 29–2909. Disponível em: <https://doi.org/10.1109/ICSA.2018.00012>. Citado 4 vezes na(s) página(s) 63, 66, 67 e 96.

FREUND, J.; RüCKER, B. *Real-Life BPMN: Using Camunda BPMN to Build Agile Process Automation*. North Charleston, SC: CreateSpace Independent Publishing Platform, 2014. ISBN 978-1507833722. Citado 3 vezes na(s) página(s) 28, 77 e 80.

FRITZSCH, J. *et al.* Towards an architecture-centric methodology for migrating to microservices. In: KRUCHTEN, P.; GREGORY, P. (Ed.). *Agile Processes in Software Engineering and Extreme Programming – Workshops*. Cham: Springer Nature Switzerland, 2024. p. 39–47. ISBN 978-3-031-48550-3. Disponível em: <https://doi.org/10.1007/978-3-031-48550-3_5>. Citado na página 64.

GIALLORENZO, S. *et al.* Lemma2jolie: A tool to generate microservice apis from domain models. *SCIENCE OF COMPUTER PROGRAMMING*, v. 228, JUN 2023. ISSN 0167-6423. Disponível em: <https://doi.org/10.1016/j.scico.2023.102956>. Citado 2 vezes na(s) página(s) 64 e 65.

GYSEL, M. *et al.* Service cutter: A systematic approach to service decomposition. In: AIELLO, M. *et al.* (Ed.). *Service-Oriented and Cloud Computing*. Cham: Springer International Publishing, 2016. p. 185–200. ISBN 978-3-319-44482-6. Disponível em: <https://doi.org/10.1007/978-3-319-44482-6_12>. Citado 4 vezes na(s) página(s) 64, 65, 68 e 92.

HENDERSON, J. C.; VENKATRAMAN, N. Strategic alignment: Leveraging information technology for transforming organizations. *IBM Systems Journal*, v. 32, n. 1, p. 4–16, 1993. Citado na página 19.

HIGHTOWER, K.; BURNS, B.; BEDA, J. *Kubernetes: Up and Running: Dive into the Future of Infrastructure*. Sebastopol, CA: O'Reilly Media, 2017. ISBN 9781491936023. Citado na página 99.

HUMBLE, J.; FARLEY, D. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Upper Saddle River, NJ: Addison-Wesley, 2010. ISBN 9780321601919. Citado 3 vezes na(s) página(s) 99, 103 e 147.

IVANCHIKJ, A.; SERBOUT, S.; PAUTASSO, C. Live process modeling with the bpmn sketch miner. *Software and Systems Modeling*, Springer, v. 21, p. 1877–1906, 2022. Disponível em: <https://doi.org/10.1007/s10270-022-01009-w>. Citado 3 vezes na(s) página(s) 77, 80 e 82.

IVAS, I. Implementation of composable enterprise in an evolutionary way through holistic business-it delivery of business initiatives: Real industry use case. In: . [s.n.], 2024. v. 1, p. 397 – 408. Cited by: 0. Disponível em: <https://doi.org/10.5220/0012728400003690>. Citado 2 vezes na(s) página(s) 68 e 92.

JACOBSON, I.; SPENCE, I.; BITTNER, K. *Use-Case 2.0: The Guide to Succeeding with Use Cases*. [S.l.]: Ivar Jacobson International, 2011. Disponível em: <https://www.ivarjacobson.com/publications>. Citado 2 vezes na(s) página(s) 78 e 80.

JOSELYNE, M. I.; BAJPAI, G.; NZANYWAYINGOMA, F. A systematic framework of application modernization to microservice based architecture. In: *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. [s.n.], 2021. p. 1–6. ISSN 2409-2983. Disponível em: <https://doi.org/10.1109/ICEET53442.2021.9659783>. Citado 4 vezes na(s) página(s) 66, 79, 84 e 86.

JOSéLYNE, M. I. *et al.* Partitioning microservices: a domain engineering approach. In: *Proceedings of the 2018 International Conference on Software Engineering in Africa*. New York, NY, USA: Association for Computing Machinery, 2018. (SEiA '18), p. 43–49. ISBN 9781450357197. Disponível em: <https://doi.org/10.1145/3195528.3195535>. Citado na página 68.

KAPFERER, S.; ZIMMERMANN, O. Domain-driven service design context modeling, model refactoring and contract generation. *Communications in Computer and Information Science*, v. 1310, p. 189 – 208, 2020. Cited by: 20. Disponível em: <https://doi.org/10.1007/978-3-030-64846-6_11>. Citado na página 88.

KAPFERER, S.; ZIMMERMANN, O. Domain-specific language and tools for strategic domain-driven design, context mapping and bounded context modeling. In: HAMMOUDI, S.; PIRES, L.; SELIC, B. (Ed.). *PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON MODEL-DRIVEN ENGINEERING AND SOFTWARE DEVELOPMENT (MODELSWARD)*. [s.n.], 2020. p. 299–306. ISBN 978-989-758-400-8. Disponível em: <https://doi.org/10.5220/0008910502990306>. Citado na página 88.

KAPFERER, S.; ZIMMERMANN, O. Domain-driven architecture modeling and rapid prototyping with context mapper. *Communications in Computer and Information Science*, v. 1361, p. 250 – 272, 2021. Cited by: 5. Disponível em: <https://doi.org/10.1007/978-3-030-67445-8_11>. Citado 11 vezes na(s) página(s) 15, 64, 67, 68, 74, 79, 80, 84, 86, 88 e 103.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. [S.l.], 2007. Citado na página 174.

LEVY, M.; LOPEZ, F.; RODRIGUEZ, D. A systematic approach for legacy systems modernization: A case study. *Journal of Software: Evolution and Process*, Wiley, v. 32, n. 8, p. e2251, 2020. Citado 2 vezes na(s) página(s) 79 e 132.

LEWIS, J.; FOWLER, M. *Microservices*: A definition of this new architectural term. 2014. On-line. <https://martinfowler.com/articles/microservices.html>, acessed on 25 de outubro de 2026. Citado 7 vezes na(s) página(s) 11, 12, 13, 26, 86, 96 e 110.

LI, C.-Y.; MA, S.-P.; LU, T.-W. Microservice migration using strangler fig pattern: A case study on the green button system. In: *2020 International Computer Symposium (ICS)*. [s.n.], 2020. p. 519–524. Disponível em: <https://doi.org/10.1109/ICS51289.2020.00107>. Citado 2 vezes na(s) página(s) 68 e 93.

MA, S.-P. *et al.* Microservice migration using strangler fig pattern and domain-driven design. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*, v. 38, n. 6, p. 1285–1303, NOV 2022. ISSN 1016-2364. Disponível em: <https://doi.org/10.6688/JISE.202211_38(6).0010>. Citado na página 93.

MUEHLEN, M. zur; RECKER, J. How much language is enough? theoretical and practical use of the business process modeling notation. In: *CAiSE 2008*. [S.l.]: Springer, 2008. Citado na página 28.

NEWMAN, S. *Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith*. Sebastopol, CA: O'Reilly Media, 2019. Citado 28 vezes na(s) página(s) 12, 13, 26, 46, 47, 66, 70, 72, 75, 83, 84, 85, 86, 87, 91, 92, 93, 94, 96, 97, 104, 108, 109, 110, 111, 142, 144 e 145.

Object Management Group. *Business Process Model and Notation (BPMN) Version 2.0.2*. 2014. <https://www.omg.org/spec/BPMN/2.0.2/>. Accessed July 2025. Citado 3 vezes na(s) página(s) 27, 77 e 82.

ÖZKAN, O.; BABUR, O.; BRAND, M. van den. Refactoring with domain-driven design in an industrial context: An action research report. *Empirical Software Engineering*, v. 28, n. 4, 2023. Cited by: 0; All Open Access, Hybrid Gold Open Access. Disponível em: <https://doi.org/10.1007/s10664-023-10310-1>. Citado 8 vezes na(s) página(s) 32, 64, 67, 74, 79, 80, 84 e 86.

P, I. V.; H, V. P. An approach to clean architecture for microservices using python. In: *2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. [s.n.], 2023. p. 1–5. Disponível em: <https://doi.org/10.1109/CSITSS60515.2023.10334229>. Citado na página 93.

PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Microservices in practice, part 1: Reality check and service design. *IEEE Software*, v. 34, n. 1, p. 91–98, 2017. Citado na página 28.

PETERSEN, K. *et al.* Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. Swindon, UK: BCS Learning & Development Ltd., 2008. (EASE'08), p. 68–77. Citado 2 vezes na(s) página(s) 34 e 172.

PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and*

Software Technology, v. 64, p. 1–18, 2015. ISSN 0950-5849. Disponível em: <https://doi.org/10.1016/j.infsof.2015.03.007>. Citado 5 vezes na(s) página(s) 14, 34, 172, 174 e 176.

PETTIGREW, A. M.; WHITTINGTON, R.; SILVA, L. S. da. Studying organizational change and development: Challenges for future research. *Academy of Management Annals*, v. 12, n. 1, p. 479–515, 2018. Citado na página 20.

PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach*. 8th. ed. [S.l.]: McGraw-Hill Education, 2014. ISBN 978-0078022128. Citado na página 11.

PRESSMAN, R. S.; MAXIM, B. R. *Software Engineering: A Practitioner's Approach*. 9th. ed. [S.l.]: McGraw-Hill Education, 2019. ISBN 978-1259872976. Citado 16 vezes na(s) página(s) 11, 22, 23, 24, 25, 28, 32, 61, 64, 65, 66, 79, 86, 92, 96 e 111.

RADEMACHER, F.; SORGALLA, J.; SACHWEH, S. Challenges of domain-driven microservice design: A model-driven perspective. *IEEE Software*, v. 35, n. 3, p. 36–43, May 2018. ISSN 1937-4194. Disponível em: <https://doi.org/10.1109/MS.2018.2141028>. Citado na página 86.

RAHMATULLOH, A. *et al.* Microservices-based iot monitoring application with a domain-driven design approach. In: *2021 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS)*. [s.n.], 2021. p. 1–8. Disponível em: <https://doi.org/10.1109/ICADEIS52521.2021.9701966>. Citado na página 89.

REIJERS, H. A.; MENDLING, J. Modelling the as-is and to-be business processes. In: *Process-Aware Information Systems*. [S.l.]: Wiley, 2011. Citado na página 28.

RICHARDSON, C. *Microservices patterns*. [S.l.]: Manning Publications Company, 2018. ISBN 9781617294549. Citado 37 vezes na(s) página(s) 12, 13, 16, 18, 19, 24, 25, 26, 32, 61, 64, 70, 75, 83, 84, 85, 86, 88, 91, 92, 93, 94, 96, 97, 103, 104, 108, 109, 110, 114, 115, 116, 125, 137, 141, 145 e 151.

SAIDI, M.; TISSAOUI, A.; FAIZ, S. A ddd approach towards automatic migration to microservices. In: *2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*. [s.n.], 2023. p. 01–06. Disponível em: <https://doi.org/10.1109/IC_ASET58101.2023.10150522>. Citado 3 vezes na(s) página(s) 66, 97 e 106.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. 2020. <https://scrumguides.org>. Acessado em: [data de acesso]. Citado na página 75.

SEVOCAB. *SEVOCAB Software and Systems Engineering Vocabulary*. 2024. Disponível em: <https://pascal.computer.org/sev_display/index.action>, Acessado em: 25 de outubro de 2026. Citado na página 35.

SILVER, B. *BPMN Method and Style: A Levels-Based Methodology for BPM Process Modeling*. Aptos, CA: Cody-Cassidy Press, 2011. ISBN 978-0982368114. Citado 3 vezes na(s) página(s) 15, 28 e 77.

SINGJAI, A.; ZDUN, U. Conformance assessment of architectural design decisions on the mapping of domain model elements to apis and api endpoints. In: *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*. [s.n.], 2022. p. 76–83. ISSN

2768-4288. Disponível em: <https://doi.org/10.1109/ICSA-C54293.2022.00058>. Citado 2 vezes na(s) página(s) 15 e 97.

SINGJAI, A. *et al.* Patterns on deriving apis and their endpoints from domain models. In: *Proceedings of the 26th European Conference on Pattern Languages of Programs*. New York, NY, USA: Association for Computing Machinery, 2022. (EuroPLoP '21). ISBN 9781450389976. Disponível em: <https://doi.org/10.1145/3489449.3489976>. Citado na página 103.

SOMMERVILLE, I. *Software Engineering*. 9th. ed. [S.l.]: Addison-Wesley, 2011. ISBN 978-0137035151. Citado 7 vezes na(s) página(s) 11, 22, 23, 24, 25, 61 e 66.

TAIBI, D.; SYSTä, K. A decomposition and metric-based evaluation framework for microservices. In: FERGUSON, D. *et al.* (Ed.). *Cloud Computing and Services Science*. Cham: Springer, 2020, (Communications in Computer and Information Science, v. 1218). p. 133–149. ISBN 978-3-030-49431-5. Citado 4 vezes na(s) página(s) 91, 92, 110 e 144.

VERNON, V. *Domain-Driven Design Distilled*. [S.l.]: Addison-Wesley Professional, 2016. ISBN 978-0134434421. Citado 12 vezes na(s) página(s) 13, 14, 29, 30, 31, 69, 70, 71, 72, 73, 75 e 84.

WANG, Y. *et al.* A reference architecture for blockchain-based traceability systems using domain-driven design and microservices. In: *2022 29th Asia-Pacific Software Engineering Conference (APSEC)*. [s.n.], 2022. p. 269–278. ISSN 2640-0715. Disponível em: <https://doi.org/10.1109/APSEC57359.2022.00039>. Citado 2 vezes na(s) página(s) 66 e 68.

WIJAYA, I. G. R.; FAJAR, A. N. A design study of microservice architecture on white label travel platform. *Journal of System and Management Sciences*, v. 13, n. 4, p. 249 – 264, 2023. Disponível em: <https://doi.org/10.33168/JSMS.2023.0415>. Citado 2 vezes na(s) página(s) 66 e 97.

WOHED, P. *et al.* On the suitability of bpmn for business process modelling. In: *Business Process Management (BPM)*. [S.l.]: Springer, 2006. Citado na página 28.

WOHLIN, C. *et al. Experimentation in Software Engineering*. Springer Publishing Company, Incorporated, 2012. ISBN 3642290434. Disponível em: <https://www.springer.com/gp/book/9783642290435>. Citado na página 42.

ZHOU, X.; XIONG, J. Automated microservice identification from design model. In: *CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications*. [s.n.], 2022. p. 1–7. Disponível em: <https://ieeexplore.ieee.org/document/9898929>. Citado 3 vezes na(s) página(s) 86, 89 e 93.