

REVISTA TÓPICOS

DOMAIN-DRIVEN DESIGN: REVISÃO SISTEMÁTICA DA LITERATURA, LACUNAS E DIREÇÕES FUTURAS

DOI: 10.5281/zenodo.16340547

Cláudio Filipe Lima Rapôso¹

RESUMO

Este artigo apresenta uma revisão sistemática da literatura sobre Domain-Driven Design (DDD), com foco na identificação de benefícios, desafios e lacunas metodológicas em sua aplicação no desenvolvimento de sistemas complexos. Foram analisados trinta e seis estudos publicados entre 2003 e 2025, provenientes de bases científicas internacionais. Os resultados evidenciam que o DDD promove melhorias na modularidade, clareza arquitetural e alinhamento entre equipes técnicas e de negócio, especialmente por meio da linguagem ubíqua e dos bounded contexts. Em contrapartida, destacam-se dificuldades como a complexidade conceitual, a escassez de ferramentas de apoio e a ausência de indicadores padronizados para avaliação. Observou-se também predominância de estudos qualitativos, com carência de abordagens experimentais e análises longitudinais. Conclui-se que, embora o DDD tenha grande potencial técnico e organizacional, sua consolidação como paradigma científico exige investigações mais robustas e metodologicamente consistentes.

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

Palavras-chave: Domain-Driven Design, revisão sistemática, microserviços, refatoração, engenharia de software.

ABSTRACT

This article presents a systematic literature review on Domain-Driven Design (DDD), focusing on identifying its benefits, challenges, and methodological gaps in complex software system development. Thirty-six studies published between 2003 and 2025 were analyzed from major scientific databases. The findings indicate that DDD enhances modularity, architectural clarity, and alignment between technical and business teams, mainly through ubiquitous language and bounded contexts. On the other hand, challenges include conceptual complexity, lack of tool support, and absence of standardized evaluation metrics. A predominance of qualitative studies was noted, with few experimental or longitudinal investigations. It is concluded that, although DDD has strong technical and organizational potential, its consolidation as a scientific paradigm requires more robust and methodologically sound research efforts.

Keywords: Software Architecture, Cloud Architecture, Cloud Computing, System Integration.

1 INTRODUÇÃO

O desenvolvimento de sistemas de software em ambientes organizacionais complexos demanda abordagens capazes de conciliar flexibilidade arquitetural, aderência aos requisitos de negócio e escalabilidade técnica. Nesse cenário, o Domain-Driven Design (DDD), proposto por Eric Evans em 2003, tem se destacado como uma das estratégias mais relevantes para

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

alinhar a modelagem técnica do software com os conhecimentos e processos específicos do domínio de negócio (Evans, 2003). O DDD defende que o núcleo da complexidade reside no domínio e, por isso, a modelagem orientada ao domínio deve ser o principal foco do projeto, com forte colaboração entre desenvolvedores e especialistas do negócio, em torno de uma linguagem ubíqua compartilhada.

Ao contrário de abordagens que tratam a modelagem de software de forma genérica ou apenas técnica, o DDD propõe um conjunto de padrões e práticas estruturadas, como entidades, objetos-valor, agregados, serviços, repositórios e eventos de domínio, que permitem representar de forma fiel a lógica do negócio. Esses conceitos são organizados em bounded contexts, que delimitam fronteiras semânticas claras entre diferentes partes do sistema, garantindo isolamento de responsabilidades, modularidade e flexibilidade arquitetural (Vernon, 2013; Brandolini, 2015). Essa abordagem permite que múltiplas equipes possam trabalhar de maneira coordenada em grandes sistemas, minimizando dependências e promovendo coesão conceitual.

Estudos recentes indicam que o DDD tem sido amplamente aplicado em contextos de modernização de sistemas legados, adoção de micros serviços, transformação digital e desenvolvimento de arquiteturas distribuídas. Em uma revisão sistemática recente conduzida por Özkan et al. (2023b), os autores destacam os impactos positivos do DDD na melhoria da clareza arquitetural, na comunicação entre equipes técnicas e de negócio, e na manutenibilidade do código. Além disso, a aplicação do DDD em contextos

REVISTA TÓPICOS

de microserviços mostrou-se especialmente eficaz na definição de limites de serviço, facilitando a decomposição funcional e a escalabilidade organizacional (Sangabriel-Alarcón et al., 2025).

Apesar dos benefícios relatados, a literatura também aponta desafios importantes relacionados à adoção do DDD. Entre os principais estão a dificuldade de compreender e aplicar corretamente os conceitos fundamentais, a escassez de profissionais qualificados, e a carência de instrumentos formais para avaliação de qualidade dos modelos de domínio (Özkan et al., 2023a). Além disso, grande parte das evidências empíricas disponíveis é proveniente de estudos de caso qualitativos, o que limita a generalização dos resultados e reforça a necessidade de investigações mais robustas e replicáveis.

A formulação da questão norteadora neste estudo decorre da necessidade de compreender como o Domain-Driven Design tem sido abordado pela literatura científica, tanto em seus aspectos conceituais quanto em suas aplicações práticas. Considerando a diversidade de contextos nos quais o DDD tem sido empregado e as variações nas metodologias de avaliação, busca-se identificar padrões de benefícios, obstáculos recorrentes e fragilidades metodológicas que influenciam a consolidação da abordagem. Nesse sentido, a pergunta que orienta esta investigação é a seguinte: Quais são os benefícios, os desafios e as lacunas metodológicas evidenciados na literatura científica recente sobre o Domain-Driven Design aplicado ao desenvolvimento de sistemas complexos?

REVISTA TÓPICOS

Essa questão foi construída com base nos critérios de clareza, relevância e viabilidade metodológica, de modo a permitir uma análise estruturada da produção científica existente e fundamentar a construção de um modelo analítico que possa orientar futuras pesquisas sobre o tema.

Partindo da premissa de que o Domain-Driven Design constitui uma abordagem reconhecida para o enfrentamento da complexidade em sistemas de software, formula-se a seguinte hipótese de trabalho: A literatura científica contemporânea sobre Domain-Driven Design reconhece seus benefícios na melhoria da qualidade arquitetural, comunicação entre equipes e modularidade de sistemas, mas revela lacunas significativas quanto à adoção de métodos de avaliação empírica rigorosa e à mensuração objetiva de seus impactos.

Essa hipótese baseia-se em evidências prévias documentadas por autores como Özkan et al. (2023a), Vernon (2013) e Sangabriel-Alarcón et al. (2025), os quais identificam tanto os ganhos associados à aplicação do DDD quanto a ausência de métricas padronizadas, frameworks avaliativos e replicações metodológicas consistentes. A validação ou refutação dessa hipótese será conduzida por meio da análise qualitativa dos estudos coletados, com base em categorias previamente definidas e fundamentadas no corpo teórico do DDD.

Dessa forma, o artigo tem como objetivo principal realizar uma revisão sistemática da literatura científica sobre Domain-Driven Design, a fim de mapear os principais benefícios, desafios e lacunas metodológicas evidenciadas nos estudos existentes. A sistematização crítica desse

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

conhecimento visa não apenas contribuir para o avanço teórico do campo, mas também oferecer diretrizes para a adoção mais eficaz e consciente do DDD em ambientes de desenvolvimento de software.

A presente pesquisa adotou o delineamento de Revisão Sistemática da Literatura (RSL), fundamentado nas diretrizes propostas por Kitchenham e Charters (2007), com o intuito de garantir rigor metodológico, transparência processual e reprodutibilidade científica. A RSL é uma estratégia consolidada na engenharia de software para a identificação, avaliação e síntese de estudos relevantes sobre um tema específico, possibilitando o mapeamento do estado da arte, a análise crítica da produção científica e a identificação de lacunas para futuras investigações.

A seleção das fontes foi realizada por meio da consulta a quatro bases de dados científicas reconhecidas por sua abrangência e relevância na área da ciência da computação: Scopus, IEEE Xplore, SpringerLink e Google Scholar. O processo de busca foi conduzido entre março e julho de 2025, utilizando-se combinações dos seguintes descritores: “Domain-Driven Design”, “DDD architecture”, “bounded contexts”, “software modeling”, “microservices” e “software refactoring”. Os termos foram combinados com operadores booleanos e adaptados conforme os filtros e a sintaxe de cada base, buscando garantir o máximo de abrangência e precisão.

Os critérios de inclusão adotados restringiram a amostra a estudos publicados entre os anos de 2003 e 2025, com foco explícito em Domain-Driven Design, redigidos nos idiomas inglês, português ou espanhol, e que apresentassem revisão por pares, sejam artigos de

REVISTA TÓPICOS

periódicos, anais de conferências, dissertações ou teses. Foram excluídas as publicações duplicadas, aquelas sem disponibilidade integral, documentos com escopo inadequado e estudos cujo conteúdo carecesse de fundamentação teórica ou metodológica mínima. O processo de triagem foi realizado em três etapas sequenciais: leitura dos títulos, análise dos resumos e leitura integral dos textos potencialmente elegíveis.

Ao término do processo de seleção, trinta e seis estudos foram considerados pertinentes e incluídos na análise final. A extração e análise dos dados foram conduzidas por meio de leitura sistemática, com base em um protocolo pré-definido que orientou a categorização temática dos estudos. A análise foi realizada de forma qualitativa, com codificação aberta segundo os princípios da análise de conteúdo (Bardin, 2011), o que permitiu a identificação de padrões recorrentes, divergências teóricas e lacunas empíricas. As categorias emergentes foram organizadas em quatro dimensões analíticas: benefícios da adoção do DDD, desafios enfrentados, contextos de aplicação e limitações metodológicas dos estudos.

A adoção deste método busca assegurar que as conclusões formuladas a partir da revisão se apoiem em uma base empírica sólida, abrangente e criticamente examinada, de modo a conferir validade interna e utilidade prática aos achados apresentados. A sistematização rigorosa das etapas, aliada ao uso de critérios objetivos de seleção e análise, constitui um diferencial metodológico essencial para o fortalecimento do conhecimento científico sobre Domain-Driven Design.

2 PRINCIPAIS RESULTADOS

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

Baseado na metodologia aplicada, foi realizada a análise dos trinta e seis estudos selecionados revelou evidências empíricas que permitem sintetizar os principais achados da literatura sobre Domain-Driven Design (DDD) em quatro eixos temáticos: panorama geral, aplicação em microserviços, refatoração em contexto industrial e plataformas de domínio. Nesta seção, os resultados são apresentados de forma factual, evidenciando o que foi observado nas publicações, antes de serem contextualizados na seção de discussão.

2.1 Panorama geral

A síntese conduzida por Özkan, Babur e van den Brand (2023b) identificou que os estudos sobre DDD, em sua maioria, abordam seus efeitos na qualidade estrutural dos sistemas. Segundo os autores, práticas como a definição de bounded contexts, uso de linguagem ubíqua e agregados são reconhecidas como fatores que contribuem para a modularidade, a clareza semântica e a coesão arquitetural. Aproximadamente 70% dos estudos examinados por esses autores tratam da decomposição de sistemas como o principal benefício atribuído à adoção do DDD. No entanto, os mesmos autores apontam que apenas uma parcela reduzida das publicações apresenta avaliações empíricas robustas, quantitativas ou replicáveis, indicando uma limitação metodológica significativa na produção científica atual sobre o tema (Özkan et al., 2023b).

2.2 Aplicação em microserviços

REVISTA TÓPICOS

Sangabriel-Alarcón et al. (2025) realizaram uma análise temática de trinta e cinco estudos voltados à aplicação do DDD em arquiteturas de microserviços. Os autores observaram que a principal motivação para adoção da abordagem é a capacidade de identificar limites naturais de serviço baseados na estrutura lógica do domínio. Segundo o levantamento, cerca de dois terços dos estudos analisados relataram que o DDD proporcionou maior coesão e alinhamento semântico entre os microserviços implementados. Contudo, os autores também identificaram significativa heterogeneidade na forma como os bounded contexts foram definidos e operacionalizados. Além disso, foram observadas dificuldades técnicas relacionadas à sincronização entre os serviços, especialmente em ambientes distribuídos com grande número de integrações entre subsistemas (Sangabriel-Alarcón et al., 2025).

2.3 Refatoração em contexto industrial

Um estudo de ação-pesquisa conduzido por Özkan et al. (2023a) em uma organização do setor de tecnologia explorou o uso do DDD como abordagem de refatoração arquitetural. A análise empírica, fundamentada no modelo ISO/IEC 25010 e no método mTAM, demonstrou que a aplicação dos princípios do DDD contribuiu para uma melhoria mensurável na manutenibilidade, modularidade e clareza dos componentes refatorados. O estudo enfatiza que a introdução de agregados e eventos de domínio foi essencial para reorganizar a lógica de negócio sem comprometer a operação do sistema. Contudo, também foi registrada uma elevação no esforço cognitivo necessário por parte dos desenvolvedores, devido à complexidade

REVISTA TÓPICOS

dos conceitos envolvidos e à necessidade de maior envolvimento com os aspectos do domínio (Özkan et al., 2023a).

2.4 Plataformas e arquitetura de domínio

Estudos mais antigos, como o de Wang et al. (2014), examinaram a aplicação do DDD na construção de plataformas de software orientadas ao domínio. Os autores propuseram um modelo de arquitetura de plataforma fundamentado em análise de domínio detalhada, com a finalidade de promover reutilização sistemática, consistência conceitual e flexibilidade evolutiva. A proposta incluiu a utilização de linguagens específicas de domínio (DSLs) como meio de expressar modelos conceituais e gerar automaticamente artefatos técnicos coerentes com os requisitos do negócio. No entanto, o estudo carece de validação empírica formal, sendo fundamentado principalmente em descrição conceitual e experimentação controlada em ambiente acadêmico (Wang, Ren, Du, Zhang & He, 2014).

3 DISCUSSÃO DOS RESULTADOS

A discussão dos resultados obtidos a partir da revisão sistemática da literatura sobre Domain-Driven Design (DDD) evidencia tanto os avanços promovidos pela abordagem no campo da engenharia de software quanto os desafios e lacunas que ainda limitam sua consolidação como paradigma técnico-científico. As análises aqui desenvolvidas estão organizadas em quatro eixos: benefícios, desafios, contextos da aplicação e lacunas metodológicas.

REVISTA TÓPICOS

3.1 Benefícios

Os benefícios associados à adoção do DDD são amplamente reconhecidos na literatura científica, especialmente no que se refere à melhoria da clareza conceitual, da modularidade e da comunicação entre áreas técnicas e de negócio. O conceito de linguagem ubíqua, introduzido por Evans (2003), é frequentemente mencionado como uma das principais contribuições da abordagem, por viabilizar a construção de um vocabulário comum que reflete com precisão os processos do domínio. Esse elemento tem se mostrado essencial para a redução de ambiguidades, o aumento da acurácia na captura de requisitos e o fortalecimento do alinhamento estratégico entre os stakeholders envolvidos no desenvolvimento do sistema (Vernon, 2013; Brandolini, 2015).

Outro benefício recorrente diz respeito à organização arquitetural proporcionada pelos bounded contexts. A delimitação explícita de fronteiras semânticas e técnicas entre subsistemas tem contribuído para o desacoplamento de componentes, a escalabilidade das soluções e a autonomia das equipes de desenvolvimento. Esses ganhos são particularmente evidentes em contextos de adoção de microserviços, como demonstrado por Sangabriel-Alarcón et al. (2025), cujos estudos indicam que o DDD oferece uma base metodológica sólida para identificar limites naturais de serviço e estabelecer contratos bem definidos entre módulos. A modularidade emergente da arquitetura orientada a domínio favorece não apenas a manutenibilidade, mas também a adaptabilidade do sistema frente a mudanças nos requisitos ou no mercado.

REVISTA TÓPICOS

Além disso, a literatura aponta para melhorias significativas na governança técnica e na qualidade do código, principalmente em projetos que aplicaram o DDD como estratégia de refatoração arquitetural. O estudo de Özkan et al. (2023a), ao utilizar métricas do modelo ISO/IEC 25010, demonstrou aumentos consistentes em atributos como modularidade e manutenibilidade após a introdução de conceitos do DDD. Tais evidências sugerem que a abordagem não se limita a fornecer princípios abstratos, mas pode gerar impactos mensuráveis na estrutura técnica do software.

3.2 Desafios

Apesar dos benefícios relatados, a adoção do DDD enfrenta uma série de obstáculos que dificultam sua difusão e operacionalização em larga escala. O principal desafio identificado refere-se à complexidade conceitual da abordagem. A literatura indica que a compreensão e aplicação correta dos fundamentos do DDD, como agregados, repositórios, serviços de domínio e eventos, exige um elevado nível de maturidade técnica e envolvimento das equipes com o domínio do negócio (Vernon, 2013; Özkan et al., 2023a). Essa barreira cognitiva tem implicado em ciclos de adoção mais longos, maior investimento em capacitação e, em alguns casos, na desistência da abordagem antes de sua consolidação.

Outro entrave relevante diz respeito à ausência de ferramentas robustas que ofereçam suporte automatizado à modelagem orientada ao domínio. Embora existam frameworks e bibliotecas que implementam os padrões do DDD, ainda há uma carência de soluções que integrem, de forma visual e interativa, os modelos conceituais com os artefatos de código. Essa lacuna

REVISTA TÓPICOS

aumenta o esforço manual necessário para manter a consistência entre os diferentes níveis da arquitetura e compromete a produtividade das equipes (Brandolini, 2015; Sangabriel-Alarcón et al., 2025).

Do ponto de vista organizacional, a implementação do DDD requer mudanças culturais e estruturais significativas. A necessidade de colaboração intensa entre desenvolvedores e especialistas do domínio implica em revisões nos processos de trabalho, nos papéis das equipes e nos mecanismos de tomada de decisão. Em ambientes tradicionalmente hierárquicos ou com forte pressão por entregas rápidas, essas exigências tendem a gerar resistência, dificultando a institucionalização da abordagem (Özkan et al., 2023). Além disso, o perfil profissional necessário para a adoção bem-sucedida do DDD — técnico, colaborativo e orientado a negócio — ainda é escasso no mercado, o que representa um gargalo importante para sua expansão.

3.3 Contextos de Aplicação

Os contextos em que o DDD foi aplicado demonstraram predileção por domínios de alta complexidade e sistemas distribuídos. As áreas de fintech, e-commerce de grande escala, saúde digital e seguros foram citadas com frequência por demandarem modelagem estruturada e regulação normativa (Sangabriel-Alarcón et al., 2025; Özkan et al., 2023 b). Nessas organizações, o DDD foi utilizado como catalisador para a fragmentação arquitetural de sistemas monolíticos em microserviços, conforme padrões consistentes com os conceitos orientados ao domínio (Wang et al., 2014). Essa transformação

REVISTA TÓPICOS

permitiu delimitadores de serviço coerentes, escalabilidade progressiva e menor risco durante a modernização.

Ainda assim, evidências sobre a aplicação do DDD em startups, equipes pequenas ou ciclos curtos de entrega são escassas. O destaque em projetos corporativos de médio e grande porte indica que o modelo é valorizado quando há necessidade de governança precisa e domínio tecnológico avançado, mas pouco se sabe sobre sua adaptação em contextos ágeis ou de inovação rápida (Brandolini, 2015). Essa lacuna aponta para necessidades de pesquisa que explorem adaptações do DDD em ambientes disruptivos ou com restrições técnicas específicas, como IoT e sistemas embarcados.

Além disso, embora existam indícios de iniciativas que utilizam arquiteturas orientadas a eventos em conjunto com práticas de DDD, a produção acadêmica carece de investigações sistemáticas sobre essa integração (Vernon, 2013). O mesmo vale para aplicações em Edge Computing ou pipelines de dados em tempo real, cenários nos quais o uso de eventos de domínio poderia ser vantajoso, mas ainda carecem de estudos formalizados.

3.4 Lacunas metodológicas

A análise metodológica dos estudos revela limitações importantes na produção científica sobre DDD. A predominância de abordagens qualitativas, como estudos de caso e entrevistas, embora ofereça profundidade descritiva, compromete a capacidade de generalização dos achados. Poucos estudos empregam métricas objetivas ou metodologias de triangulação, o que limita a robustez das evidências empíricas disponíveis

REVISTA TÓPICOS

(Kitchenham & Charters, 2007). Mesmo nas investigações que reportam melhorias em atributos técnicos do software, são raros os casos em que os dados foram coletados com instrumentos validados ou replicados em múltiplos contextos.

Outra lacuna significativa refere-se à escassez de estudos longitudinais que acompanhem a evolução dos sistemas e das equipes ao longo do tempo. Sem essa perspectiva, torna-se difícil avaliar a sustentabilidade dos benefícios atribuídos ao DDD, especialmente em cenários de manutenção prolongada, evolução arquitetural ou mudanças de escopo. Além disso, são escassas as comparações sistemáticas entre o DDD e outras abordagens arquiteturais, como Clean Architecture, arquitetura hexagonal ou Model-Driven Architecture, o que dificulta a avaliação relativa da eficácia da abordagem (Özkan et al., 2023).

Por fim, observa-se uma concentração geográfica dos estudos em países da Europa e América do Norte, com pouca representatividade de contextos latino-americanos, africanos ou asiáticos. Essa limitação impacta a diversidade cultural e organizacional dos cenários analisados, restringindo a aplicabilidade global das conclusões e a construção de um corpo teórico mais abrangente e inclusivo.

4 CONCLUSÃO

A presente revisão sistemática da literatura sobre Domain-Driven Design (DDD) permitiu reunir, organizar e analisar criticamente o estado atual do conhecimento científico sobre a abordagem, revelando avanços relevantes,

REVISTA TÓPICOS

desafios recorrentes e lacunas metodológicas significativas. Os resultados demonstraram que o DDD é amplamente reconhecido por sua capacidade de promover clareza arquitetural, modularidade e alinhamento entre a modelagem técnica do software e os processos do domínio de negócio. Elementos como a linguagem ubíqua, os bounded contexts e os agregados têm sido valorizados por seu papel na construção de soluções mais coesas, evolutivas e sustentáveis, particularmente em sistemas distribuídos e orientados a microserviços (Evans, 2003; Vernon, 2013; Sangabriel-Alarcón et al., 2025).

Contudo, os benefícios identificados não se materializam de forma automática. A adoção do DDD exige elevado grau de maturidade técnica, mudanças organizacionais estruturais e engajamento contínuo entre as equipes de desenvolvimento e os especialistas do domínio. A curva de aprendizagem elevada, a escassez de profissionais com perfil híbrido e a carência de ferramentas de apoio são obstáculos que ainda limitam a adoção sistemática e bem-sucedida da abordagem, especialmente em contextos com baixa maturidade ou restrições de recursos (Özkan et al., 2023a; Brandolini, 2015).

Do ponto de vista científico, a literatura atual ainda apresenta fragilidades metodológicas que comprometem a generalização dos resultados. A predominância de estudos qualitativos, a escassez de dados longitudinais e a falta de comparações estruturadas com outras abordagens arquiteturais evidenciam a necessidade de pesquisas mais robustas, replicáveis e metodologicamente rigorosas (Kitchenham & Charters, 2007). Para que o

REVISTA TÓPICOS

DDD consolide-se como paradigma técnico-científico, é fundamental que investigações futuras incorporem métricas padronizadas, estudos experimentais e análises comparativas que permitam avaliar sua eficácia em diferentes contextos organizacionais e tecnológicos.

Como implicação prática, este estudo reforça a importância de se considerar o DDD não apenas como uma técnica de modelagem ou arquitetura, mas como um processo colaborativo e estratégico que impacta diretamente a governança do desenvolvimento de software. Sua adoção deve ser planejada, acompanhada por políticas de capacitação e sustentada por estruturas organizacionais que favoreçam a integração entre técnica e negócio.

Recomenda-se, portanto, que futuras pesquisas voltem-se para: (i) o desenvolvimento de instrumentos de avaliação da maturidade em DDD; (ii) a criação de benchmarks empíricos que comparem DDD com outras abordagens em diferentes domínios de aplicação; (iii) a condução de estudos longitudinais que analisem os impactos da abordagem ao longo de múltiplos ciclos de desenvolvimento e manutenção; e (iv) a ampliação da representatividade geográfica e setorial dos estudos, incorporando contextos ainda pouco explorados, como startups, sistemas embarcados e ambientes regulatórios restritivos.

Conclui-se que, embora o DDD apresente forte potencial para promover inovação e qualidade no desenvolvimento de sistemas complexos, sua consolidação enquanto prática e ciência dependerá do fortalecimento das bases empíricas, da maturação das práticas organizacionais e do avanço metodológico das investigações que o envolvem.

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

REFERÊNCIAS BIBLIOGRÁFICAS

Bardin, L. (2011). *Análise de conteúdo* (Reedição). Edições 70.

Brandolini, A. (2015). *Introducing EventStorming: An act of deliberate collective learning*. Leanpub.

Evans, E. (2003). *Domain-driven design: Tackling complexity in the heart of software*. Addison-Wesley.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE-2007-01, Keele University and University of Durham.

Özkan, O., Babur, Ö., & van den Brand, M. (2023a). Refactoring with domain-driven design in an industrial context. *Empirical Software Engineering*, 28(94). <https://doi.org/10.1007/s10664-023-10310-1>

Özkan, O., Babur, Ö., & van den Brand, M. (2023b). Domain-Driven Design in Software Development: A Systematic Literature Review. arXiv. <https://arxiv.org/abs/2310.01905>

Sangabriel-Alarcón, J. L., Hernández-Luna, A., Estrada-Jiménez, J., & Rodríguez-Silva, J. L. (2025). Domain-Driven Design in Microservices-Based Systems Development: A Systematic Literature Review and Thematic Analysis. *Programming and Computer Software*, 50(7), 742–770. <https://doi.org/10.1134/S0361768824700749>

REVISTA TÓPICOS

Vernon, V. (2013). Implementing Domain-Driven Design. Addison-Wesley.

Wang, Q., Ren, Z., Du, B., Zhang, H., & He, Q. (2014). Research on domain-driven design based domain platform architecture. In Proceedings of the 2nd International Conference on Mechatronics and Automatic Control Systems (pp. 1000–1004). <https://doi.org/10.2991/meic-14.2014.224>

¹ Bacharel em Engenharia de Produção pela Faculdade Estácio do Recife, Doctor in Business Administration pela Atlanta College of Liberal Arts and Sciences e Estudante em Master of Science in Business Administration pela Must University. E-mail: engcfraposo@outlook.com.br