

APLICAÇÃO DO DOMAIN-DRIVEN DESIGN PARA A ARQUITETURA DE UM SISTEMA PARA MARKETPLACE

NATHAN AUGUSTO DIAS RODRIGUES¹, PATRICK DA SILVA NICEZI²,
CRISTIANE YAE MI IMAMURA³

¹Graduado em Engenharia de Software, Centro Universitário das Faculdades Associadas de Ensino – UNIFAE, Campus São João da Boa Vista, nathanaugustorodrigues@hotmail.com.

²Graduado em Engenharia de Software, Centro Universitário das Faculdades Associadas de Ensino – UNIFAE, Campus São João da Boa Vista, patrick0806@hotmail.com.

³Mestre em Ciências pela USP – São Carlos. Professora dos cursos de Engenharia de Computação e Engenharia de Software, Centro Universitário das Faculdades Associadas de Ensino – UNIFAE, Campus São João da Boa Vista, cristiane@fac.br. Área de conhecimento (Tabela CNPq): 1.03.03.02-2 Engenharia de Software

RESUMO: A medida que o campo tecnológico ganha relevância no contexto comercial, evidencia-se a necessidade de determinar uma arquitetura de software que proporcione adequadamente a representação dos componentes de um sistema de software e suas relações, permitindo o controle sobre o desenvolvimento e implementação do mesmo. Neste trabalho, é utilizado o padrão de modelagem *Domain-Driven Design* (DDD), e através dele foi possível perceber melhor os diferentes contextos de domínios da aplicação, as entidades de cada contexto, além de outros artefatos, para a determinação de uma arquitetura de microserviços. Os resultados apresentados aqui se referem a aplicação do DDD para a definição da arquitetura como um todo e são detalhados os artefatos para o contexto de carrinho de compras, uma vez que esse é um dos principais serviços de um marketplace.

PALAVRAS-CHAVE: arquitetura de software; comércio eletrônico; sistema web; microserviços.

APPLYING DOMAIN-DRIVEN DESIGN FOR THE ARCHITECTURE OF A MARKETPLACE SYSTEM

ABSTRACT: As the technological field gains relevance in the commercial context, the need to determine a software architecture that adequately provides the representation of the components of a software system and its relationships, allowing control over its development and implementation, becomes evident. In this work, the Domain-Driven Design (DDD) modeling pattern is used, and through it was possible to better understand the different contexts of the application domains, the entities of each context, in addition to other artifacts, for the determination of an architecture of microservices. The results presented here refer to the application of DDD to define the architecture as a whole and detail the artifacts for the shopping cart context, since this is one of the main services of a marketplace.

KEYWORDS: software architecture; e-commerce; web system; microservices.

INTRODUÇÃO

Conforme o setor tecnológico evolui e com o crescente acesso a internet, diversos segmentos de mercado perceberam a possibilidade de aumentar seus lucros oferecendo seus serviços e produtos por meio de plataformas online (MENDONÇA, 2016). Esses fatores marcaram o surgimento do e-commerce (comércio eletrônico) e alavancaram o fluxo de vendas. Segundo uma pesquisa recente, o e-commerce cresceu 12% em vendas online no primeiro semestre de 2019 (CHINAGLIA, 2019).

Com o crescimento da necessidade de sistemas de software para marketplace e em função de sua natureza cada vez mais complexa, exige-se para esse tipo de software o uso de uma arquitetura

bem planejada em termos de desenvolvimento e funcionalidades. Uma arquitetura de software bem definida, permite aos stakeholders gerenciar detalhes dos componentes do sistema (CORRÊA, 2017).

Entre os padrões de arquitetura existentes, a arquitetura de microsserviços vem ganhando destaque pela sua orientação a modularização e escalabilidade (AMARAL, 2018). Entretanto, existem diferentes estratégias para determinar quais serviços e artefatos farão parte desse tipo de arquitetura, e muitos trabalhos não chegam a detalhar como empregam cada estratégia.

O objetivo desse trabalho é mostrar como é possível determinar a arquitetura de um software de marketplace, evidenciando parte dos artefatos encontrados. Tendo em vista a importância do domínio da aplicação, foi escolhido aplicar o padrão denominado *Domain-Driven Design* (DDD).

MATERIAL E MÉTODOS

Em Microsoft (2019a), foi possível encontrar um padrão de recomendação de projeto que muitas empresas tem usado para descobrir os microsserviços. Esse padrão é conhecido como *Domain-Driven Design*, desenvolvido por Evans (2003), e é empregado com base na camada de Domínio (*Domain*) que é responsável por representar conceitos, situações e regras de negócio. Então, para seu uso é necessário isolar os aspectos do domínio da aplicação.

O DDD é dividido em duas fases, a estratégica e a tática. A fase estratégica consiste na definição da estrutura em grande escala do sistema e assegurar que a arquitetura mantenha-se concentrada em capacidades comerciais, e a tática contém um conjunto de padrões de design para criar o modelo de domínio, a fim de manter os microsserviços coesos e com flexibilidade para acoplamento (MICROSOFT, 2019a).

Um dos objetivos da fase estratégica é a definição de Contextos Limitados. Um mesmo software pode ter diferentes contextos de aplicação, de acordo com uma regra de negócio a ser aplicada, dependendo do setor ou departamento que o usará. Cada área independente de negócio da aplicação pode ser vista como um contexto limitado e cada contexto limitado está correlacionado a um microsserviço (MICROSOFT, 2018).

Durante a fase tática, para cada contexto limitado encontrado na fase estratégica, prevê-se a definição de outros padrões como Entidades, Objetos de Valor e Serviços (MICROSOFT, 2019b):

- Entidades são objetos com identidade exclusiva que persiste ao longo do tempo.
- Objetos de Valor são objetos sem identidade conceitual.
- Agregações são conjuntos de Entidades relacionadas por associação.
- Serviços são objetos que implementam uma lógica sem conter nenhum estado, e neste contexto estão relacionados aos microsserviços.

Assim, para definir a arquitetura desejada, foram aplicados os passos de acordo com a Figura 1.

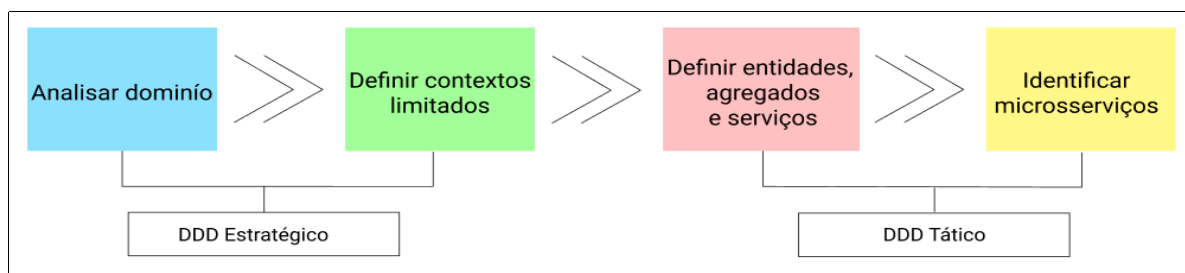


FIGURA 1. Etapas para a definição dos microsserviços com DDD (MICROSOFT, 2019a).

Conforme a Figura 1 ilustra, nesse trabalho foram realizadas quatro etapas, sendo as duas primeiras para promover o DDD estratégico e as duas últimas o DDD tático:

Etapa 1 – Foram elencados os requisitos do sistema e esboçada uma visão panorâmica do sistema de marketplace de forma informal, apenas para entender as regras do negócio e a visão de cada usuário ou departamento.

Etapa 2 – Foram mapeadas as funções de negócio e suas conexões, modelado o domínio corporativo de modo a criar um modelo de domínio, definidos os contextos e seus respectivos limites propriamente dito. A etapa 1 e 2 são essenciais para filtrar e organizar os dados de conhecimento de domínio e fornecer uma linguagem comum entre os stakeholders (MICROSOFT, 2019a).

Etapa 3 – Para cada contexto limitado encontrado na Etapa 2, definiram-se as Entidades, Objetos de Valor, Agregações e Serviços.

Etapa 4 – Avaliou-se e identificou-se cada um dos microsserviços, realizando os seguintes passos, de acordo com Microsoft (2019c).

1. Definiu-se o microsserviço por contexto limitado, uma vez que a funcionalidade do mesmo não deve abranger mais de um contexto.
2. Foram examinados os agregados, pois eles são fortes candidatos a microsserviços.
3. Foram avaliados os serviços de domínios, pois são bons candidatos a microsserviços.
4. Foi analisado cada microsserviço e se ele contemplava as seguintes características:
 - Ter uma única responsabilidade.
 - Não fazer nenhuma chamada de conversa entre outros serviços.
 - Ser pequeno o suficiente para ser criado por uma pequena equipe trabalhando de forma independente.
 - Não haver nenhuma interdependência que exija que seja implantado em sincronia com outro.
 - Poder evoluir de forma independente.
 - Verificar se os limites do serviço não criarão problemas de consistência ou integridade de dados.

RESULTADOS E DISCUSSÃO

Na Etapa 1 do DDD estratégico, foi feita uma análise dos requisitos funcionais para o sistema de marketplace proposto, descrevendo informalmente o domínio geral da aplicação, de modo a compreender que existem entidades que participam desse software com diferentes visões, dependendo do contexto no qual é visto. Por exemplo, para Produto se o contexto for o gerenciamento de Catálogo, as informações necessárias são maiores em número e em detalhes. Para o Catálogo, o preço de custo do produto importa, enquanto no contexto do Carrinho de Compras o preço de custo não precisa aparecer na modelagem. A documentação completa dessa visão informal do sistema pode ser vista em um documento disponível no Github (RODRIGUES; NICEZI; IMAMURA, 2020a).

Com a descrição informal do negócio, iniciou-se a Etapa 2 do DDD estratégico e nela foi possível perceber os diversos contextos de domínio da aplicação de marketplace. Em cada contexto uma parte do negócio fica segregada das demais, embora os domínios precisem se comunicar, dependendo de suas entidades e serviços que possuírem. Os contextos limitados e com quem eles se comunicam podem ser vistos na Figura 2.

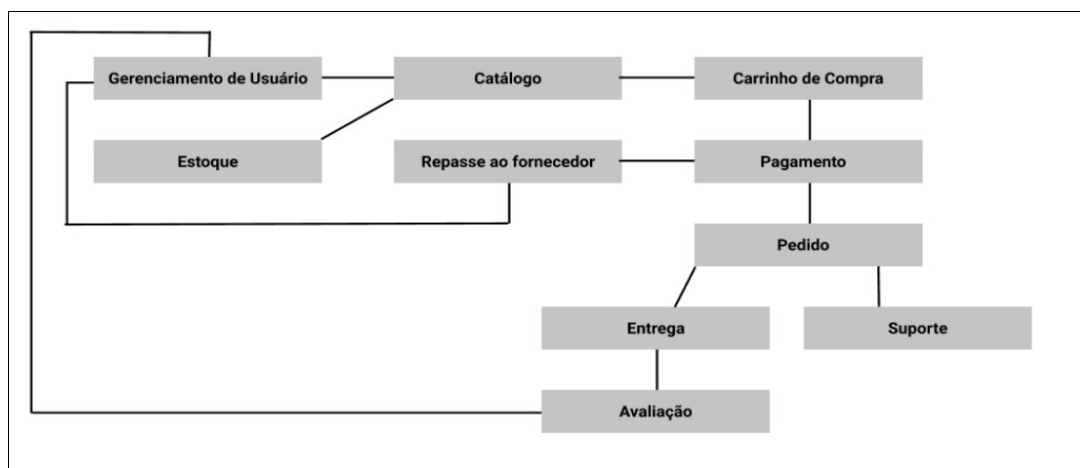


FIGURA 2. Contextos Limitados e seus relacionamentos (RODRIGUES; NICEZI; IMAMURA, 2020c).

Os contextos limitados devem trabalhar como um conjunto para a realização de uma tarefa maior, dessa forma os limites de um contexto podem ser próximos de outros contextos. Além disso, cada contexto limitado pode ser visto como um sistema dentro do sistema de marketplace, representando um certo subdomínio da aplicação maior.

Para cada contexto limitado encontrado, foi aplicado o DDD tático para definir entidades, objetos de valor, agregações e serviços, correspondendo a execução da Etapa 3. Na Figura 3, podem ser vistos os artefatos encontrados para o contexto do Carrinho de Compras. Esse contexto foi

demonstrado por se comunicar com o contexto de Catálogo de Produtos e por ser o contexto que dispara a realização de um possível Pedido. Para o software proposto, um pedido depende da realização de um Carrinho e de um Pagamento.

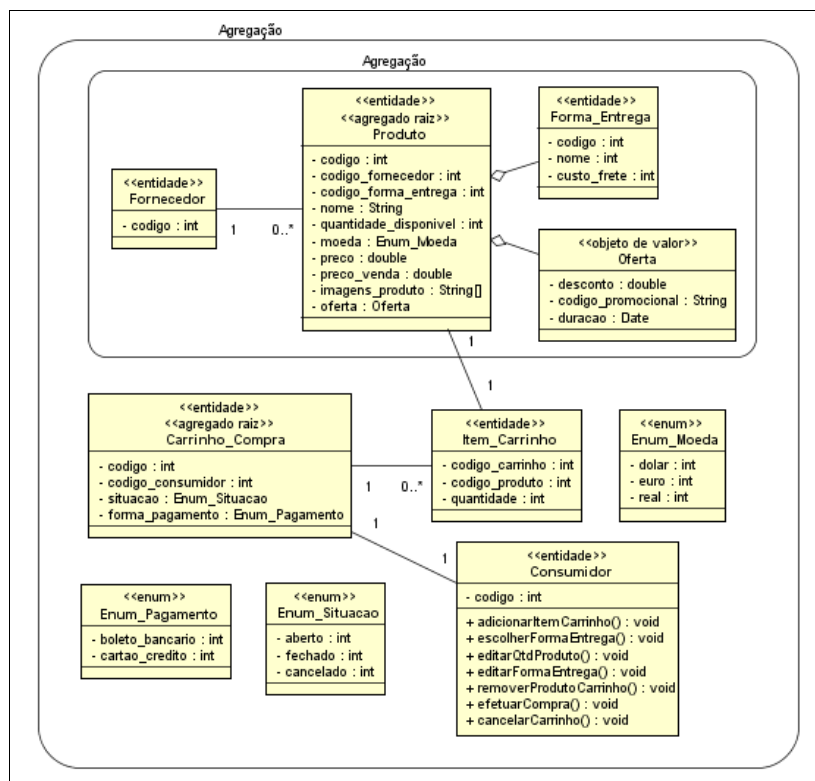


FIGURA 3. DDD tático aplicado ao contexto de Gerenciamento do Carrinho de Compra (RODRIGUES; NICEZI; IMAMURA, 2020b).

Para o Carrinho de Compras, foram encontradas as seguintes entidades: Consumidor, Fornecedor, Produto, Formas de Entrega, Carrinho de Compra e Item do Carrinho. Oferta é o objeto de valor. Produto é agregado com Oferta, com Forma de Entrega e Fornecedor e, nessa primeira agregação, Produto é a raiz. Além disso, há uma agregação maior com todas as entidades, na qual a raiz é o Carrinho de Compra, pois a partir dele são acessados os Itens do Carrinho e demais objetos. Nesse escopo há um serviço de domínio que deve controlar os itens incluídos no carrinho e validá-los, e este serviço foi chamado de Carrinho.

Com base nos resultados anteriores, foi realizada a Etapa 4:

1. Cada contexto foi analisado e foi constatado que suas funcionalidades não abrangem os limites de outros contextos.
2. Foram avaliadas as agregações, mas como elas faziam parte de outra agregação maior, conclui-se que acabavam mapeando e correlacionando as entidades totais do domínio onde se encontravam, portanto, dessa análise não resultou microsserviços. Como o caso do Carrinho de Compras que resultou em um único microsserviço.
3. Cada serviço de domínio foi eleito como microsserviço. Assim foram elencados os microsserviços: Gerenciamento do Usuário, Repasse ao Fornecedor, Avaliação, Suporte, Entrega, Pedido, Pagamento, Carrinho, Catálogo e Estoque.
4. Cada microsserviço citado foi analisado e em função de Gerenciamento de usuário possuir mais que uma responsabilidade que poderia contemplar as características dos microsserviços, surgiu o último microsserviço definido para essa arquitetura: Controle de Acesso ao Usuário.

De posse dos resultados, foi possível compreender melhor a complexidade dos sistemas de marketplaces e que cada microsserviço pode ser implementado de fato independentemente e com uma tecnologia distinta, evoluindo independente de outros serviços. Entretanto, foi observado, que uma vez que cada microsserviço possui sua própria infraestrutura, as alterações de suas entidades podem refletir em comunicações para a atualização dos dados da entidade em outros contextos, que possuem uma visão diferente da mesma entidade.

CONCLUSÕES

Aqui foram mostrados os passos seguidos desde a identificação do contexto do sistema até a definição dos microsserviços. Entretanto, é necessário salientar que tais microsserviços podem acabar sendo alterados quando postos em prática frente aos adicionais padrões de projetos a serem usados para a implementação do software. Assim, para melhor definição dessa arquitetura, como trabalhos futuros é sugerida a integração dos aspectos das outras camadas da arquitetura, além da de negócio.

Vale salientar também que devido aos diferentes aspectos que um sistema dessa natureza pode ter, a relação de microsserviços de um outro sistema para marketplace pode diferir daquela apresentada nesse artigo. Diante da dificuldade inicial para identificar os microsserviços de um sistema de marketplace, com base nos resultados demonstrados é possível concluir que o DDD fornece um meio para analisar os diferentes aspectos da aplicação e permite o levantamento dos serviços e microsserviços que podem compor a arquitetura. Foi possível perceber que a aplicação desse padrão de modelo direciona naturalmente a arquitetura para os microsserviços.

REFERÊNCIAS

AMARAL, José Diogo Coelho. A evolução das arquiteturas monolíticas para as arquiteturas baseadas em microserviços. 2018. Dissertação (Mestrado em Informática) – ISEP – Instituto Superior de Engenharia do Porto. Disponível em: <<http://hdl.handle.net/10400.22/11920>>. Acesso em: 08 jun 2020.

CHINAGLIA, Rafael. E-commerce cresce 12% no primeiro semestre de 2019, segundo 40º webshoppers. 2019. Disponível em: <<https://www.ecommercebrasil.com.br/noticias/e-commerce-cresce-12-por-cento-webshoppers-i-e-commerce-brasil/3/>>. Acesso em: 29 mar 2020.

CORRÊA, Andreiwd Sheffer. Uma arquitetura de referência colaborativa para estruturação de dados abertos governamentais. 2017. Tese (Doutorado em Sistemas Digitais) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2017. Disponível em: <<https://teses.usp.br/teses/disponiveis/3/3141/tde-28062017-101825/pt-br.php>>. Acesso em: 05 maio 2020. doi:10.11606/T.3.2017.tde-28062017-101825.

EVANS, Eric. Domain-driven design: tackling complexity in the heart of software. 1. ed. Addison-Wesley Professional, 2003.

MENDONÇA, Herbert Garcia de. E-commerce. Revista IPTEC, [S.l.], v. 4, n. 2, p. 240-251, dec. 2016. ISSN 2318-9851. Disponível em: <<http://www6.uninove.br/ojs/journaliji/index.php/iptec/article/view/68/146>>. Acesso em: 05 maio 2020. doi:10.5585/iptec.v4i2.68.

MICROSOFT. Projetar um microsserviço orientado a DDD. 2018. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>>. Acesso em: 03 maio 2020.

MICROSOFT. Usando a análise de domínio para modelar microserviços. 2019a. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/model/domain-analysis>>. Acesso em: 03 maio 2020.

MICROSOFT. Usando DDD tático para projetar microsserviços. 2019b. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/model/tactical-ddd>>. Acesso em: 03 maio 2020.

MICROSOFT. Identificando limites de microsserviço. 2019c. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/model/microservice-boundaries>>. Acesso em: 03 maio 2020.

RODRIGUES, Nathan Augusto Dias; NICEZI, Patrick Da Silva; IMAMURA, Cristiane Yae Mi. Descrição informal dos requisitos funcionais de um sistema para marketplace. 2020a. Disponível em: <<https://github.com/CONICT2020-Nathan-Patrick-Cristiane/Anexos>>. Acesso em: 17 ago 2020.

RODRIGUES, Nathan Augusto Dias; NICEZI, Patrick Da Silva; IMAMURA, Cristiane Yae Mi. Descrição informal do contexto dos domínios limitados e aplicação do DDD tático. 2020b. Disponível em: <<https://github.com/CONICT2020-Nathan-Patrick-Cristiane/Anexos>>. Acesso em: 17 ago 2020.

RODRIGUES, Nathan Augusto Dias; NICEZI, Patrick Da Silva; IMAMURA, Cristiane Yae Mi. Do contexto de domínio até a definição dos microsserviços para uma arquitetura de um sistema de marketplace. 2020c. Trabalho de Conclusão de Curso – Centro Universitário das Faculdades Associadas de Ensino – UNIFAE, São João Da Boa Vista, 2020.