

DO CONTEXTO DE DOMÍNIO ATÉ A DEFINIÇÃO DOS MICROSSERVIÇOS PARA UMA ARQUITETURA DE UM SISTEMA DE MARKETPLACE

Nathan Augusto Dias Rodrigues – UNIFAE – nathanaugustorodrigues@hotmail.com

Patrick Da Silva Nicesi – UNIFAE – patrickk0806@hotmail.com

Cristiane Yae Mi Imamura – UNIFAE – cimamura@gmail.com

Resumo

Conforme o campo tecnológico avança, a necessidade de atender não apenas processos de negócios, mas também conceitos das camadas de apresentação e tecnologia ganham propriedade e igual relevância. Então, faz-se necessário primar pela definição da arquitetura de software, que permite representar os componentes de um sistema e seus relacionamentos, garantindo assim o controle sobre o desenvolvimento e implementação do mesmo. Por meio do padrão de modelagem Domain Drive Design (DDD), foi possível analisar o domínio de uma aplicação de marketplace e definir os microsserviços de sua arquitetura. Neste trabalho são mostrados os passos resultantes da aplicação do DDD, sendo possível compreender que a arquitetura de microsserviços se torna bastante interessante e quase inerentemente escolhida, se comparada à tradicional arquitetura monolítica, em aplicações de grande porte.

Palavras-chave: Arquitetura de Software; E-commerce; Sistema Web; Microservices; Domain Drive Design.

1- Introdução

Com o avanço da tecnologia e a crescente popularização da internet, se tornou comum e de fácil acesso da população a utilização dos recursos de tecnologia da informação, com isso o mercado dos mais diversos segmentos viu possibilidade de ampliar seus lucros oferecendo serviços e produtos por meio de plataformas online (MENDONÇA, 2016).

Desta forma, foi marcado o surgimento do e-commerce (comércio eletrônico) levando ao crescimento do fluxo de vendas. No ano de 2018, o e-commerce cresceu 24% em escala mundial, faturando 2,9 trilhões de dolares em vendas (BAGATINI, 2019). E segundo uma pesquisa recente, o e-commerce cresceu 12% no primeiro semestre de 2019 (CHINAGLIA, 2019).

Assim houve uma maior complexidade dos sistemas, gerando a necessidade de uma arquitetura de software bem planejada tanto em desenvolvimento quanto para garantir aspectos importantes de funcionalidade (AMARAL, 2018).

Uma arquitetura de software bem definida é de extrema importância, pois permite aos stakeholders uma visualização detalhada dos componentes do sistema, auxiliando nas tomadas de decisões que envolvem o mesmo. Diante disso, é possível controlar o desenvolvimento do sistema, baseando-se nos aspectos de funcionalidade, disponibilidade, manutenção, requisição de usuários e entre outros atributos de qualidade (CORRÊA, 2017).

Há alguns padrões de arquitetura que se destacam, como a arquitetura tradicional monolítica e a arquitetura de microsserviços. Recentemente a arquitetura de microsserviços tem ganhado destaque e com ela as aplicações devem ser desmembradas em componentes separados mínimos e independentes que vão colaborar para realizar as mesmas tarefas, em oposição à arquitetura monolítica, na qual a aplicação é construída como um único bloco centralizado (AMARAL, 2018).

Tendo em vista a importância da arquitetura de software e a tendência do padrão de microsserviços e também da popularização crescente dos softwares de e-commerce, o foco do presente trabalho é a definição de uma arquitetura de microsserviços para marketplace, mostrando os passos seguidos e os respectivos resultados alcançados, desde a fase de análise do domínio até a descoberta dos microsserviços.

2- Justificativa

Conforme o crescente avanço tecnológico ganha propriedade e relevância no contexto social e comercial, muitos softwares e sistemas acabam se tornando obsoletos ou obtêm um grande grau de complexidade conforme escalam em tamanho (HAIDER et al., 2018). Isso se deve a constante necessidade do sistema ser adaptável diante das exigências do mercado de TI, padrões de qualidade e ciclo de desenvolvimento. Como também, ao crescimento do número de consumidores informados, que exigem produtos de software entregues em tempos cada vez menores, com mais funcionalidade e maior qualidade em termos de construção e apresentação (AMARAL, 2018).

Ainda segundo AMARAL (2018), sistemas de software devem ser flexíveis, pois é de suma importância que os mesmos sejam de fácil articulação entre as equipes no processo de desenvolvimento. Então, se faz necessária a utilização do conceito de arquitetura de software,

que permite representar os componentes de um sistema e seu relacionamento com o ambiente ao qual será instaurado, garantindo assim controle sobre o desenvolvimento e implementação do mesmo.

A arquitetura de software fornece a estrutura para que a mudança seja possível, pois sem a facilidade para suportar mudanças não há agilidade. Outro ponto a ressaltar é que a relação de dificuldade e causa, segue o princípio de Pareto. Onde em 80% da dificuldade encontrada no software se encontra em 20% dos elementos do sistema e que ao construir uma arquitetura bem definida fica evidente quais são esses 20% (JÚNIOR, 2018).

Entretanto, definir uma arquitetura de software consome muito tempo e planejamento, e exige uma comunicação constante entre os stakeholders. Então uma abordagem popularmente utilizada, consiste em usar uma arquitetura de referência voltada para o sistema que se pretende implementar e a partir dela realizar modificações na estrutura para que se atenda a necessidade do sistema em questão (AULKEMEIER et al., 2016; CORRÊA, 2017).

Para atender a finalidade de construir um sistema para comércio eletrônico, existem muitos modelos de referência disponíveis em artigos científicos, que estão relacionados com a criação de um e-commerce. Entre esses modelos, alguns a serem citados são: o modelo H de Becker para varejo, que fornece uma arquitetura de referência específica de domínio para empresas de varejo (J. Becker e R. Schutte, 2007, apud AULKEMEIER et al., 2016); o E-MEMO de Frank, que contém processos e estratégias de negócios de referência para e-commerce (U. Frank, 2004, apud AULKEMEIER et al., 2016); e o de Burt e Sparks que investiga o impacto do comércio eletrônico no processo de varejo (S. Burt e L. Sparks, 2003, apud AULKEMEIER et al., 2016).

Porém, poucos focam em um escopo global e normalmente analisam as regras de negócios de um comércio eletrônico, tecnologias a serem usadas ou partes da estrutura de funcionamento da loja, como financeiro, logística de vendas, atendimento e suporte ao cliente (AULKEMEIER et al., 2016).

Entre os modelos de referência, pode-se citar o de Aulkemeier et al. (2016), o qual se distingue ao abranger não apenas processos de negócios, mas também conceitos das camadas de apresentação e tecnologia, mas que dista do trabalho proposto aqui por não se tratar de uma arquitetura de microserviços. A arquitetura baseada em microserviços é uma abordagem à modularização de software, na qual o sistema será segmentado em contextos menores e

implementado por serviços pequenos com seus próprios dados e a escalabilidade é efetuada de forma independente, tornando-o mais resiliente a erros (AMARAL, 2018).

Diante do exposto o presente trabalho busca contribuir ao definir uma arquitetura que atenda aspectos gerais para a construção de um sistema de marketplace, de forma a mostrar os passos necessários para chegar à arquitetura objetivo, já que grande parte dos projetos relacionados a e-commerce apenas demonstram uma pequena parte da arquitetura e não demonstram os passos para chegar até ela.

3- Objetivos

3.1- Objetivo Geral

Devido à grande notoriedade do padrão da arquitetura de microsserviços e da tendência cada vez maior da construção de aplicações para e-commerce, objetiva-se definir uma arquitetura que possa ser usada para um software de marketplace, de forma que sejam mostrados os passos necessários para a determinação dos microsserviços.

3.2- Objetivos Específicos

- Elencar um padrão de metodologia de projeto a ser seguido e que possa colaborar para a definição da arquitetura.
- Entender o domínio da aplicação e seus respectivos requisitos.
- Analisar as vantagens e desvantagens da arquitetura de microsserviços.

4- Desenvolvimento

4.1- Referencial Teórico

4.1.1- E-commerce

E-commerce é um modelo de negócio que se fortaleceu com o avanço da internet, facilitando o processo de compra e venda, e permitindo com que as empresas e fornecedores disponibilizem seus serviços ou produtos de forma online (MENDONÇA, 2016). Dessa maneira os consumidores podem realizar suas transações comerciais a qualquer momento

independente do horário ou lugar, possibilitando aos mesmos conhecer, comparar e comprar produtos diferentes da maneira mais conveniente possível (MAGALHÃES, 2015).

Os e-commerces podem, ser caracterizados em diferentes tipos, abordando um determinado aspecto das relações comerciais, entre eles se destacam os seguintes: *Business To Business* (B2B), *Business To Customers* (B2C), *Business To Governments* (B2G), *Governments To Governments* (G2G) e *Customers To Customers* (C2C).

O tipo B2B é voltado para transações entre empresas, essa relação acontece por meio de redes privadas compartilhadas entre elas, e se caracteriza por vendas diretas e suporte comercial, pois sua finalidade é detalhar seu produto de venda e a sua relação com a empresa negociante (TARAS; VICHINHESKI, 2016).

Já o B2C ocorre quando as empresas vendem seus produtos ou serviços diretamente aos consumidores, desse modo a internet age como um mediador nas transações, e os consumidores por sua vez compartilham suas informações e realizam transações (AZEVEDO, 2015).

Na categoria B2G, o espaço virtual é destinado a empresa e o governo para realizarem determinado projeto, coordenar reuniões online e gerenciar o seu progresso. Nesse tipo também está presente o aluguel de plataformas e de bancos de dados com fins de uso estritamente governamental. Por sua vez a G2G, é uma categoria onde se tem relação apenas entre governos, os quais podem usar desse meio para realizar transações com governos locais e internacionais (TARA; VICHINHESKI, 2016).

E a categoria C2C, tem como objetivo o contato entre consumidores para que os mesmos façam negociações sem interferência direta de uma empresa, usando de uma plataforma e-commerce específica para fazer a troca de informações e realizar as transações (TARA; VICHINHESKI, 2016).

Existem diversos softwares da categoria B2C, que podem ser classificados como marketplace. Um marketplace é uma plataforma de e-commerce que agrupa lojas virtuais com o objetivo de comercializar produtos e serviços, e por estarem sediados na internet não atingem apenas consumidores regionais, mas também possibilitam a negociação e comercialização global (TAKAHATA, 2017).

Para fazer a análise do domínio do marketplace que seria projetado neste trabalho, levou-se em consideração a plataforma de e-commerce da Amazon, por sua abrangência no mercado global e também por ter iniciado a sua atuação no mercado brasileiro. Foram estudadas também a plataforma de comércio online do Magazine Luiza e Mercado Livre. O Magazine

Luiza foi a terceira maior empresa de e-commerce brasileiro por três anos consecutivos desde 2016 (SBVC, 2018). Já o Mercado Livre foi estudado devido a sua característica de negócio permitir que pessoas físicas também se tornem vendedores de produtos em sua plataforma.

A Amazon é uma das líderes no setor global do comércio eletrônico e tem sede nos Estados Unidos, ela iniciou como uma empresa de comércio eletrônico, mas na realidade seus modelos de negócios se expandiram além das atividades baseadas na Internet e se tornaram mais complexos, e hoje ela é uma empresa de comércio eletrônico que atende o modelo B2B, B2C e C2C (WU; GEREFFI, 2019). Entretanto, o B2C continua sendo seu principal negócio, onde a opção de mercadorias se estende dos livros a quase tudo, como milhões de eletrônicos, roupas, acessórios, autopeças, artigos para a casa, produtos de beleza, brinquedos, entre outros produtos.

O Magazine Luiza foi uma das empresas pioneiras no Brasil a se lançar no comércio eletrônico e a oferecer uma plataforma de marketplace. Nesse sentido, de acordo com Vidigal e Benfeita (2019) em 2011, o Magazine Luiza traz ao mercado uma iniciativa de social commerce, chamado Magazine Você, em que empreendedores podem criar sua própria loja virtual sem a necessidade de grandes investimentos e ganham comissões por isso. A logística da Magazine Luiza é uma operação multicanal gerido por um backoffice único com 12 centros de distribuição. As vendas podem ser realizadas pelo marketplace, loja convencionais, lojas virtuais, e-commerce, televendas, aplicativo ou pelo Magazine Você.

Já o Mercado Livre se diferencia do modelo tradicional de e-commerce em vários pontos, mas um dos grandes diferenciais está na descentralização do estoque e possibilidade de ampliar a variedade de produtos. O marketplace virtual funciona semelhante a um físico, no qual o fornecedor entra em contato com o marketplace verifica as regras de adesão e demais taxas, e então abre sua conta e então adiciona ao seu perfil os seus produtos a vender. No caso do consumidor, ele acessa o marketplace pela plataforma online, e então pode encontrar os mais variados segmentos de produtos e serviços em um único lugar, agregando valor a experiência de compra (TAKAHATA, 2017).

4.1.2- Arquitetura de Software

A arquitetura de software consiste na abstração de um sistema de software, de modo a descrever e detalhar a organização fundamental desse sistema, a fim de permitir uma visualização e controle sobre a complexidade do mesmo. Desta forma é possível representar os

componentes de um sistema e seu relacionamento com o ambiente ao qual será ou foi instaurado, permitindo também uma avaliação prévia antes da implementação do sistema (CORRÊA, 2017).

Ainda segundo Corrêa, (2017), uma arquitetura bem definida permite ao sistema ser escalável, e garante suporte a eventuais mudanças no sistema tais como customizações e configurações, e além de tudo possibilita uma fácil comunicação entre os stakeholders do sistema.

4.1.2.1- Padrões de Arquitetura

Baseado em Maldonado et al. (2002), um padrão descreve uma solução para um determinado problema que acontece com frequência durante o desenvolvimento do software, podendo conter um conjunto de informações a respeito da estrutura e do raciocínio por trás da comprovação da solução, bem como o contexto em que acontece o problema e o conjunto de suas repercussões. Padrões de arquitetura de software, podem se referir a diferentes graus de abstração e por meio do seu uso é possível estabelecer um vocabulário comum para a comunicação entre projetistas, garantindo uniformidade na estrutura do software como também servir de base para projetos complexos serem construídos.

Durante a fase de pesquisas, dois padrões de arquitetura se destacaram: o monolítico e o de microserviços.

Na arquitetura monolítica, a aplicação é construída sobre um único bloco executável lógico, normalmente é composta por uma única unidade de software possuindo três partes: uma interface de interação com o usuário, uma base de dados e uma aplicação servidor para o processamento de requisições, atualização de dados e lógica de domínio (SOUZA; PELISSARI, 2017). Ainda de acordo com Souza e Pelissari (2017), o desenvolvimento de uma aplicação monolítica corresponde ao modelo natural de implementação de um sistema, e nela toda a lógica é manipulada através de um processo singular, possibilitando o uso de características básicas da linguagem de desenvolvimento utilizada.

A solução monolítica possui vários benefícios como simplicidade para desenvolver, implantar e dimensionar, entretanto quando a aplicação se tornar grande e a equipe de desenvolvimento crescer as desvantagens se tornam cada vez mais evidentes e significativas. Entre as desvantagens podem se destacar a dificuldade para os desenvolvedores entenderem e

modificarem os códigos, maior tempo para iniciar a aplicação devido ao tamanho da mesma, dificuldade para atualizar componentes sendo necessário reimplementar a aplicação inteira e o sistema pode se tornar constantemente dependente da tecnologia em que foi construído, dificultando a utilização de novas tecnologias (RICHARDSON, 2020).

Já a arquitetura de microsserviços, é uma abordagem para modularizar o software, onde o conceito de desenvolvimento aplicado é o de decompor os modelos de domínio comercial do sistema, a fim de implementá-los em um conjunto de serviços pequenos e isolados, dessa forma os transformando em contextos menores, consistentes e limitados (AMARAL, 2018). Sendo composto por vários serviços, esse padrão possibilita que a escalabilidade seja executada de maneira independente e com grande resiliência a erros, devido a cada serviço possuir dados únicos.

Para o sistema permanecer unido, flexível e adaptável a comunicação é realizada por meio da resposta de serviços (nesse contexto denominados de microsserviços) a funções de negócio ou de apoio, assim garantindo uma maior flexibilidade e capacidade de adaptação ao sistema (AMARAL, 2018). Esse padrão de arquitetura permite ao sistema atuar como um conjunto de serviços colaborativos fracamente acoplados, assim o tornando altamente manutenível e testável. E também permite a equipe de desenvolvimento trabalhar de maneira independente em seus serviços, proporcionando alta produtividade devido a possibilidade de alocar pequenas equipes para o desenvolvimento (RICHARDSON, 2020).

Entretanto, ainda de acordo com Richardson (2020), essa solução possui suas próprias desvantagens, como a complexidade de se implementar um sistema distribuído, tratar as falhas parciais, testar as interações entre serviços, grande grau de coordenação necessária entre as equipes e complexidade para implantação e gerenciamento de um sistema composto por muitos serviços diferentes.

Em virtude do porte de um sistema marketplace ser considerado grande e frente às vantagens citadas, em detrimento das desvantagens, foi escolhido o padrão de arquitetura de microsserviços como escopo deste trabalho.

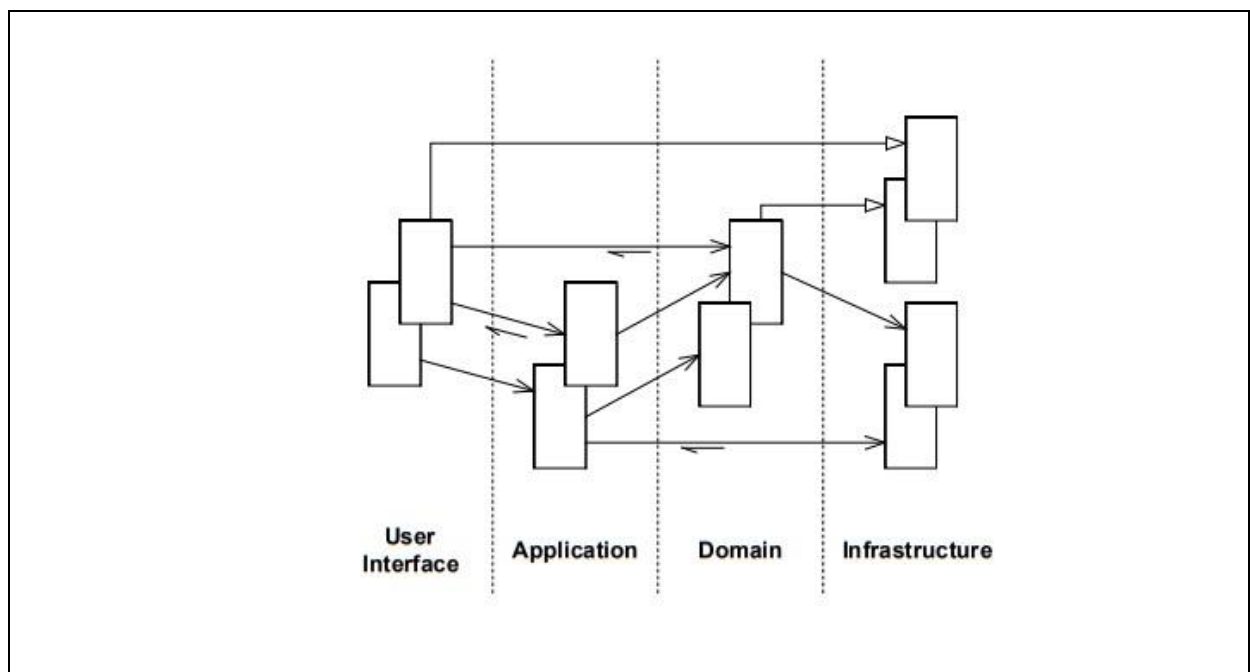
4.1.3- Domain Driven Design

Muitas publicações a respeito de microsserviços explanam a respeito das características fundamentais desse padrão, mas não expõem uma metodologia para definir esse tipo de

arquitetura. Com uma pesquisa mais aprofundada, sobre empresas que oferecem serviços e produtos para a composição de um arquitetura de microserviços, como em Microsoft (2019a), foi possível chegar ao padrão de recomendações de projeto que muitas empresas têm usado para guiar a descoberta dos microserviços. Esse padrão é chamado de *Domain Drive Design*, mais citado como DDD.

O nome desse padrão de modelagem foi designado por Evans (2003), e segundo ele a maioria das arquiteturas de sucesso usa alguma versão de arquitetura conceitual de quatro camadas, como a ilustrada na Figura 1.

Figura 1 – Arquitetura em Camadas.



Fonte: Evans (2003).

A camada de Interface do Usuário (*User Interface*) é responsável por mostrar as informações ao usuário e interpretar os comandos dos usuários. A camada de Aplicação (*Application*) coordena as tarefas e delega as colaborações dos objetos de domínio na camada seguinte. A camada de Domínio (*Domain*) é responsável por representar os conceitos, situações e regras do negócio. Já a camada de Infraestrutura (*Infrastructure*) prove as capacidades técnicas que dão suporte às camadas anteriores, enviando mensagens para a aplicação, persistindo o domínio, etc.

No DDD o modelo é proposto com base na camada de Domínio. Assim, para aplicação desse modelo é necessário isolar a implementação do domínio.

O DDD possui duas fases, a estratégica onde é definida a estrutura em grande escala do sistema e garante que a arquitetura permaneça concentrada em capacidades comerciais, e a tática onde é estabelecido um conjunto de padrões de design para criar o modelo de domínio, desse modo tornando os microserviços coesos e com flexibilidade para acoplamento (MICROSOFT, 2019a).

Defendendo uma modelagem com base na realidade dos negócios, o DDD sugere muitos conceitos e padrões de projeto a serem usados e, na fase estratégica, um deles é a definição de Contextos Limitados. Cada área independente de negócio da aplicação pode ser vista como um contexto limitado e cada contexto limitado está correlacionado a um microserviço (MICROSOFT, 2018).

Durante a fase tática, esse modelo prevê a definição de outros padrões como: Entidades, Objetos de Valor, Agregações, Serviços, Fábrica, Repositórios e Módulos. Segundo 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 microserviços.

Evans (2003) define que:

- Fábrica encapsula o conhecimento necessário para criar objetos complexos.
- Módulo também conhecidos como Pacotes, são agrupamentos de conceitos do modelo, e não de código.
- Repositório representa todos os objetos de um certo tipo de conjunto conceitual.

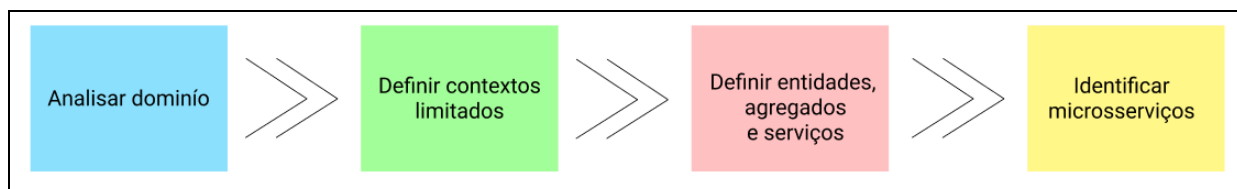
4.2- Materiais e Métodos

Para que fosse possível definir os microserviços de forma coesa, foi escolhido aplicar o modelo *Domain Driven Design* em função de sua estrutura clara, e por permitir a partir de

suas recomendações a definição dos passos iniciais necessários para uma arquitetura de microserviços.

Aplicação do DDD consistiu na aplicação dos passos ilustrados na Figura 2.

Figura 2 – Etapas para definição dos Microserviços com DDD.



Fonte: Microsoft (2019a).

A análise do domínio e a definição dos contextos limitados fazem parte do DDD estratégico. Para contemplá-las foi necessário obter uma visão panorâmica do sistema de marketplace, por meio de um mapeamento de todas as funções de negócios e suas respectivas conexões. E então foi modelado o domínio corporativo a fim de criar um modelo de domínio. Essas etapas são importantes para filtrar e organizar os dados de conhecimento de domínio e fornecer uma linguagem comum para desenvolvedores, especialistas de domínio, arquitetos de software e outros stakeholders (MICROSOFT, 2019a).

Após mapear o domínio corporativo e definir os contextos limitados para o modelo de domínio, foi iniciada a fase tática do DDD com a definição das Entidades, Objetos de Valor, Agregações e Serviços. E após a definição dos Serviços, foi avaliada a granularidade e as características dos mesmos para a definição dos microserviços.

Para isso, foram seguidos os seguintes passos, de acordo com Microsoft (2019c):

1. Definiu-se o microserviço 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 microserviços.
3. Foram avaliados os serviços de domínios, pois são bons candidatos a microserviços.
4. Foi analisado cada microserviço 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.
- Pode evoluir de forma independente.
- Os limites do serviço não criarão problemas de consistência ou integridade de dados.

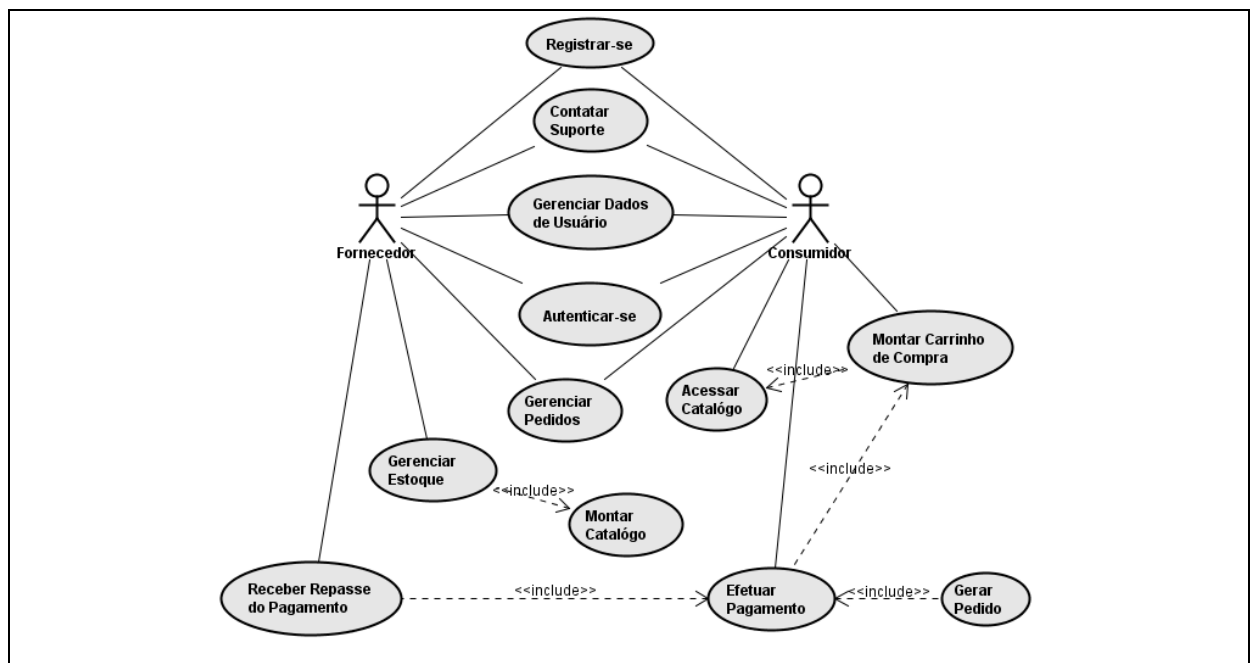
4.3- Resultados e Discussões

4.3.1- Definição da Arquitetura

Para realizar a análise do domínio da aplicação do marketplace, ou seja a primeira etapa do DDD estratégico, foram investigadas as atividades que os clientes podem fazer através do site da Amazon, Magazine Luiza, Mercado Livre e quais ações que os vendedores frequentemente fazem nessa última plataforma. Assim, como primeira etapa foi feita a análise dos requisitos funcionais para um sistema dessa natureza.

Dessa primeira etapa, tem-se como resultado um diagrama de Caso de Uso exposto na Figura 3, através do qual podem ser observadas as ações mínimas a serem suportadas pelo sistema, para os dois tipos de atores principais: Consumidor e Fornecedor.

Figura 3 – Escopo mínimo de requisitos do sistema de marketplace.



Fonte: Próprio Autor.

Outra saída dessa etapa foi a construção de uma documentação que descreve informalmente o domínio geral da aplicação, através da lógica e regras de negócio envolvidas em cada caso de uso. A seguir na Tabela 1, encontram-se alguns trechos dessa documentação.

Tabela 1 – Parte do domínio informalmente descrito.

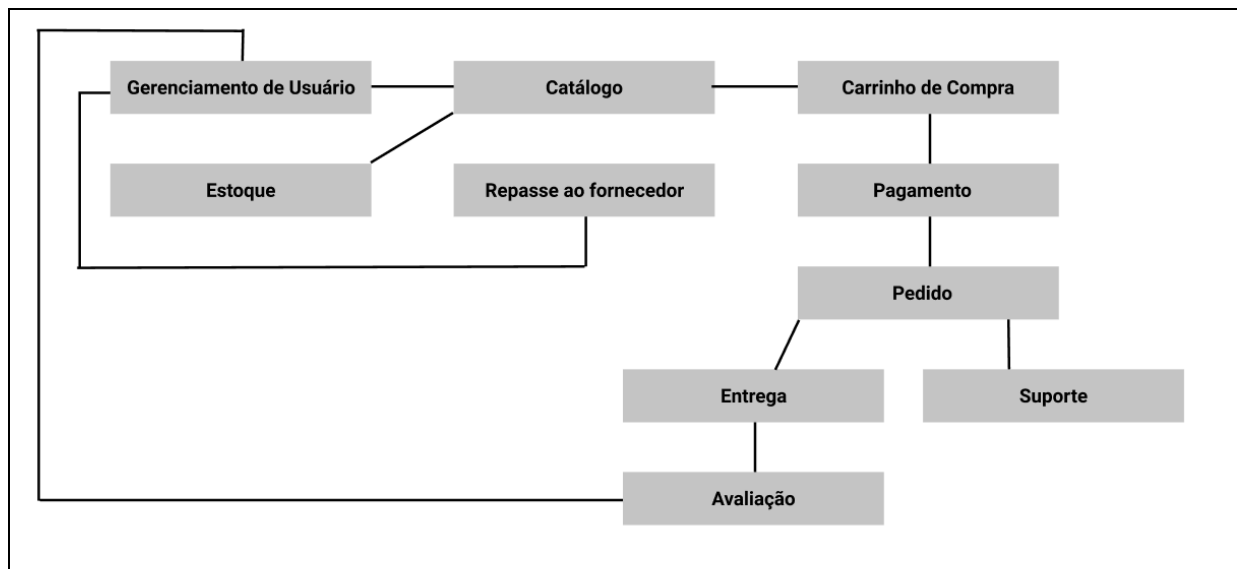
1 – Ponto de Vista do Consumidor	2 – Ponto de Vista do Fornecedor
1.1. O cliente poderá visualizar os produtos disponíveis no catálogo (visualizar: preço, imagem, tamanho, nome do produto, de qual fornecedor ele é, quem vai entregar).	2.1. O fornecedor poderá registrar seus produtos na loja virtual do aplicativo (nome do produto, valor, quantidade, categoria, preço base, preço de venda, tamanho, imagem, ficha técnica (marca, peso, cor, validade... depende de categoria)).
1.2. O cliente poderá se registrar (cadastrar-se) no sistema (nome, e-mail, senha, CPF, telefone, endereço, data de nascimento).	2.2. O fornecedor poderá se registrar (cadastrar-se) no sistema (CNPJ, nome fantasia, e-mail, senha, telefone, endereço).

Fonte: Próprio Autor.

Essa descrição informal do negócio foi a entrada para a segunda etapa do DDD estratégico. Como pode ser percebido dependendo do contexto algumas entidades são necessárias, outras não. Ou dependendo do contexto do domínio da aplicação, uma mesma entidade pode apresentar variações.

Observando os itens 1.1 e 2.1 da Tabela 1, é possível concluir que a entidade Produto apresenta atributos diferentes dependendo se o contexto for o cliente visualizar os produtos do catálogo ou se o contexto for o fornecedor registrar seus produtos para integrarem o catálogo de produtos a serem vendidos. Dessa forma, fazendo a análise de que entidades eram necessárias para um certo contexto, e também se havia entidades que mudavam de atributos dependendo do contexto, foi realizada a segunda fase do DDD estratégico. Dessa fase, o resultado foi a definição dos contextos limitados apresentados na Figura 4.

Figura 4 – Contextos Limitados.

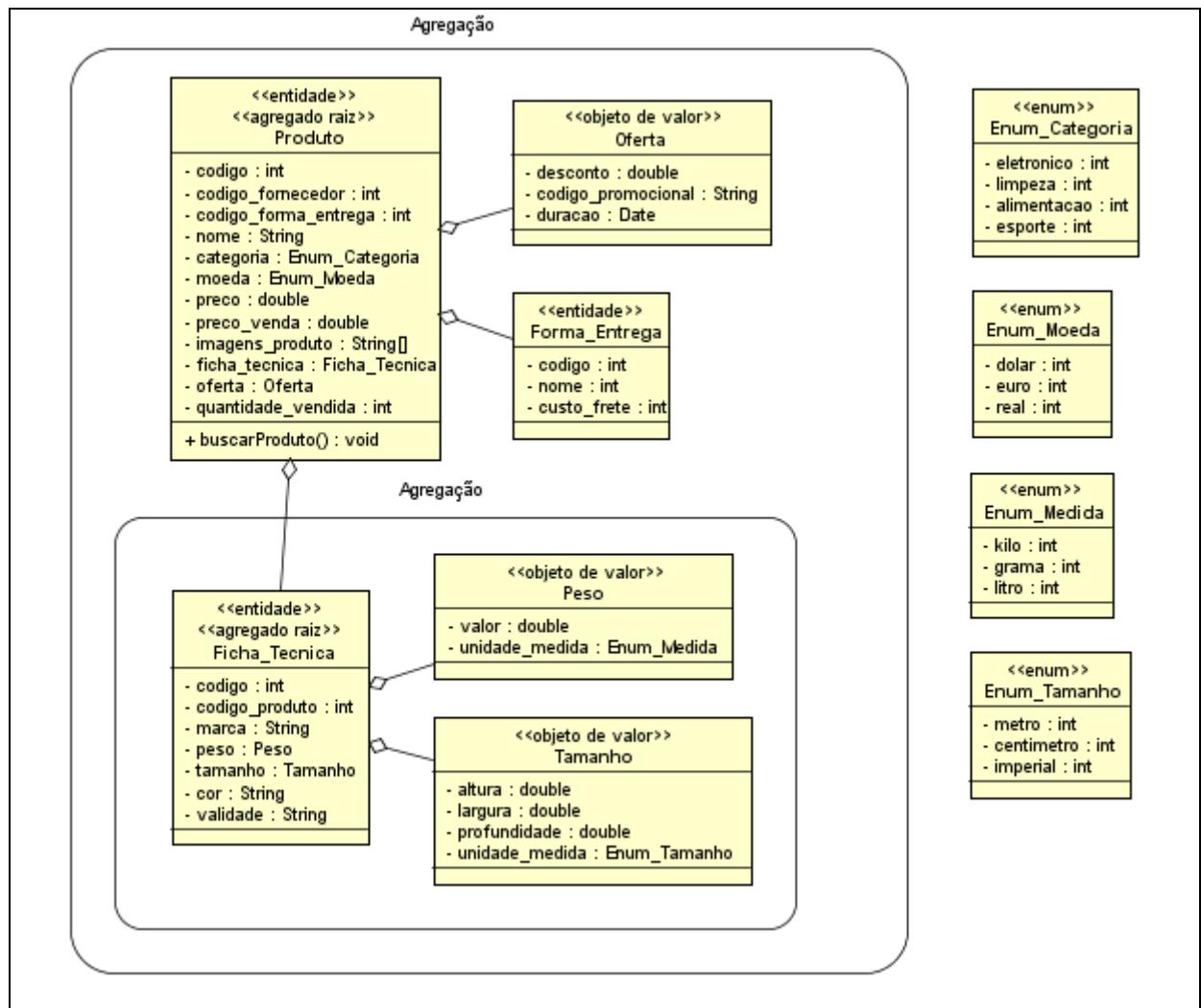


Fonte: Próprio Autor.

Cada contexto limitado encontrado tem a complexidade para ser encarado como um sistema dentro do sistema de marketplace, representando um subdomínio específico da aplicação maior. Para cada um deles, foi aplicado o DDD tático para definir entidades, objetos de valor, agregações e serviços de domínio.

Tendo em vista que um sistema de marketplace é dependente da qualidade do catálogo de produtos que oferece, aqui é apresentado o resultado da aplicação do DDD tático sobre o contexto limitado de Gerenciamento de Catálogo, na Figura 5. Na Figura 6 são mostradas as entidades, agregações e objetos de valor do Gerenciamento de Carrinho de Compra, pela sua proximidade com o contexto de Pedido. Já na Figura 7, são apresentados os resultados sobre o contexto de Gerenciamento de Usuários, para enriquecer a discussão da definição dos microserviços.

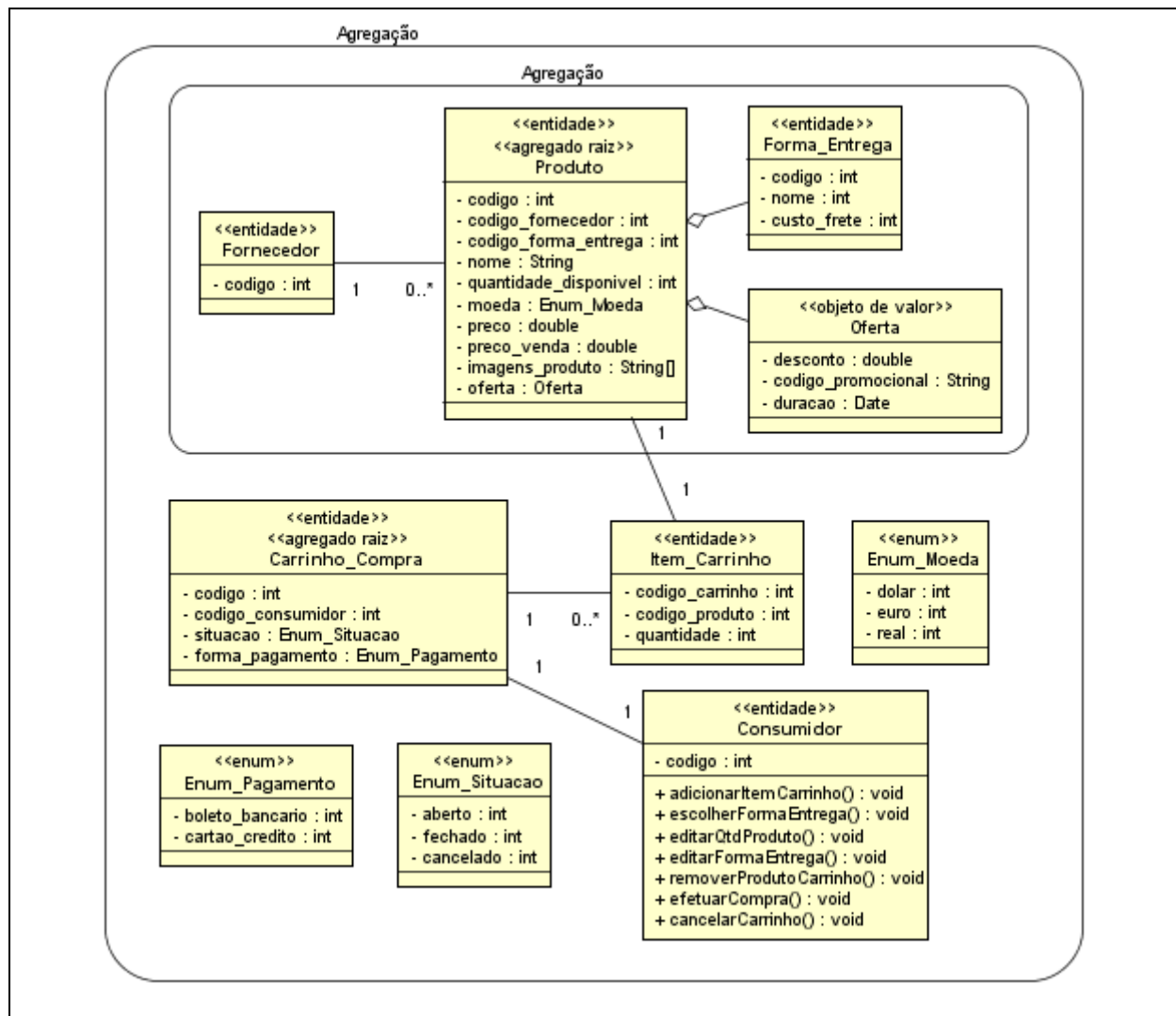
Figura 5 – DDD Tático sobre o Gerenciamento de Catálogo.



Fonte: Próprio Autor.

No Contexto de Gerenciamento de Catálogo, foram identificadas as entidades: Produto, Forma de Entrega e Ficha Técnica. Dentro desse cenário os objetos de valor incluem: Oferta, Peso e Tamanho. Como primeira agregação tem-se o conjunto formado por Ficha Técnica, Peso e Tamanho. Entretanto, analisando as dependências, Produto, Ficha Técnica, Oferta e Forma de Entrega fazem parte de uma agregação maior, cuja raiz é Produto que controlará o acesso às demais entidades dentro dos limites desse domínio. Para esse domínio limitado, foi definido o serviço de domínio de Catálogo que inclui ações como realizar buscas dinâmicas de produtos e categorizar os produtos a serem exibidos.

Figura 6 – DDD Tático aplicado ao contexto de Gerenciamento de Carrinho de Compra.



Fonte: Próprio Autor.

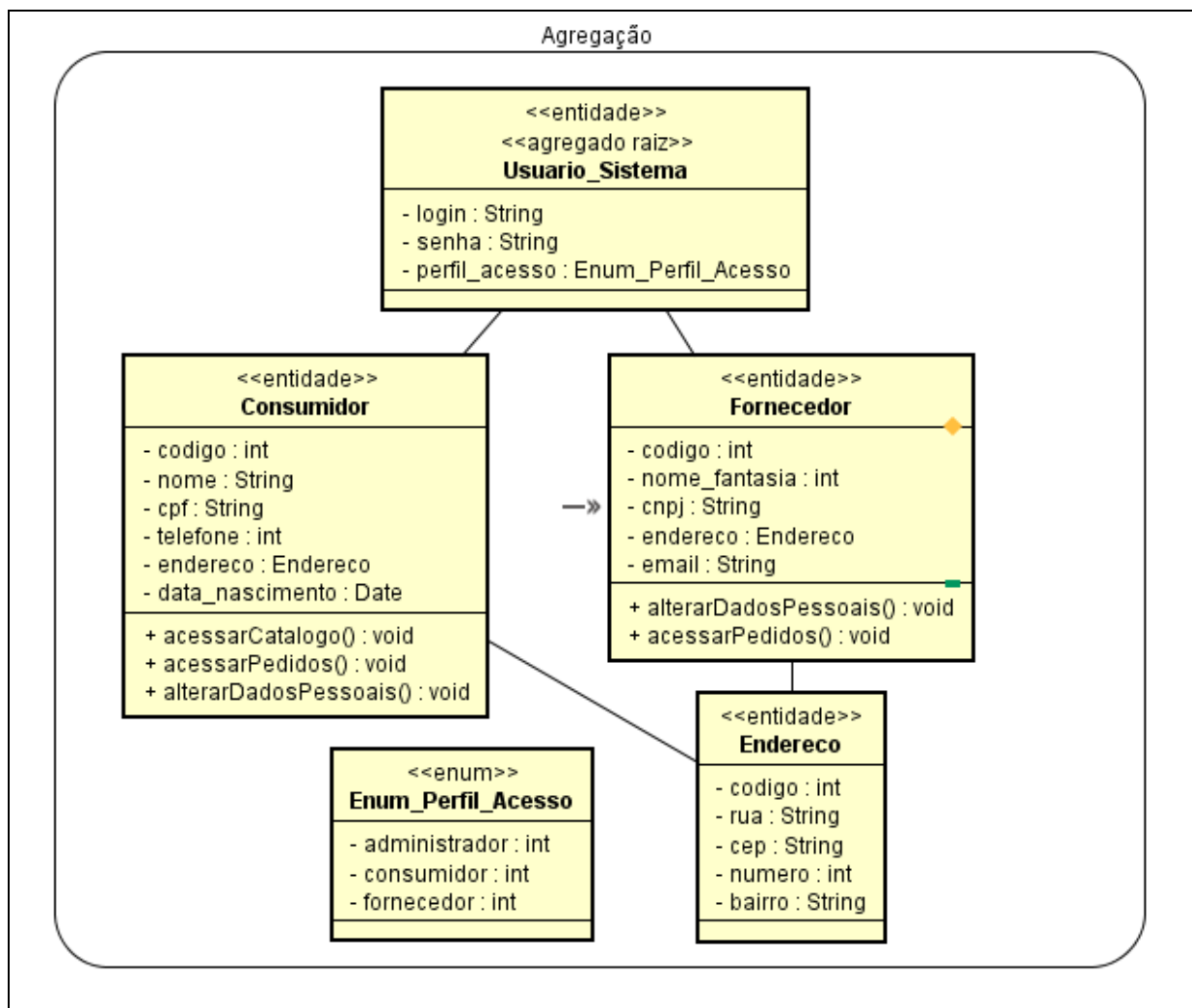
No contexto de Gerenciamento de Carrinho de Compras, foram consideradas as seguintes entidades: Consumidor, Fornecedor, Produto, Forma de Entrega, Carrinho de Compra e Item do Carrinho. Oferta é o objeto de valor. Produto é agregado com Oferta, com Forma de Entrega e Fornecedor e, nessa primeira agregação, Produto é a raiz. Além disso, há uma agregação maior com todas as entidades, mas 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.

Comparando a Figura 5 com a Figura 6 é possível verificar como a entidade Produto se distingue nesses contextos. Além disso, percebe-se que a entidade Consumidor não precisa

fazer parte do domínio de Gerenciamento de Catálogo, evidenciando assim a independência entre esses domínios.

Figura 7 – DDD Tático para o Gerenciamento de Usuário.



Fonte: Próprio Autor.

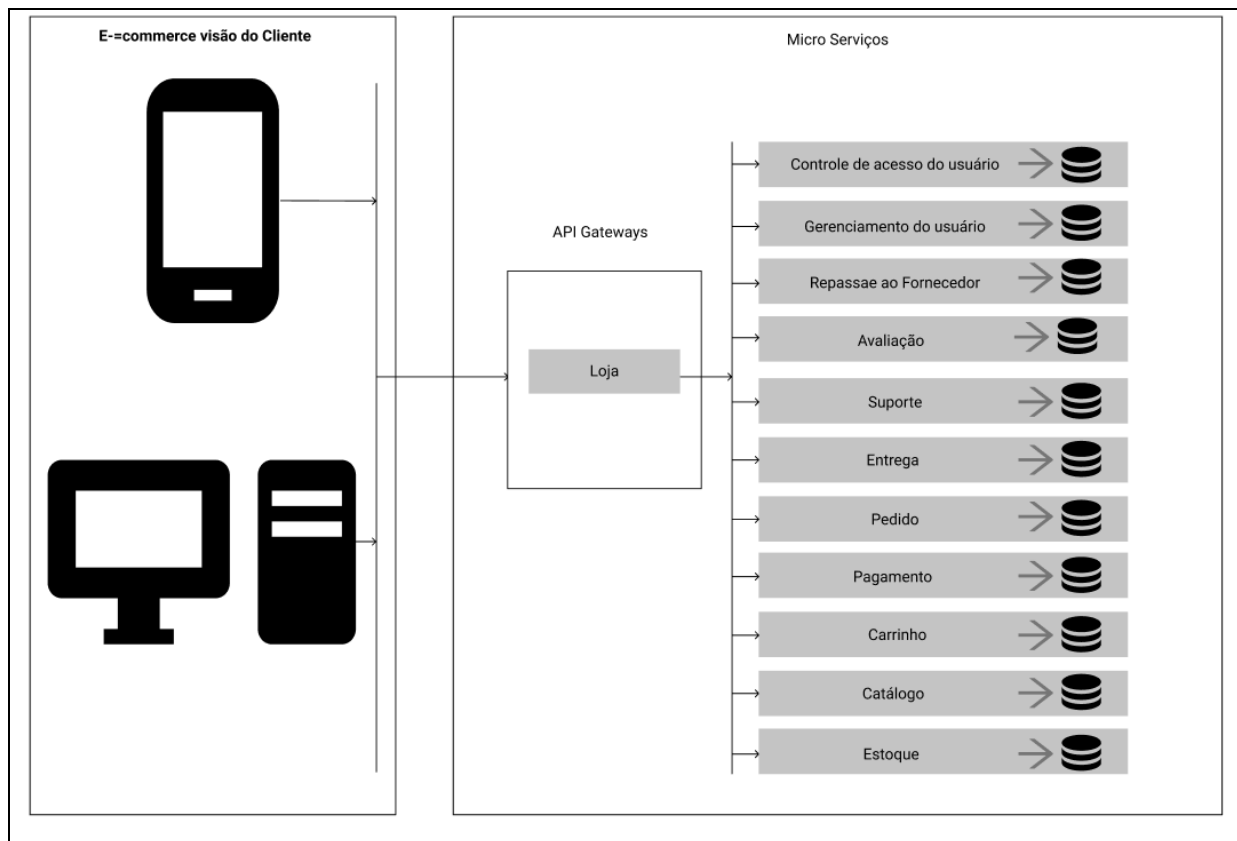
Para o domínio limitado de Gerenciamento de Usuários, foram designadas as seguintes entidades: Fornecedor, Consumidor, Endereço e Usuário do Sistema. Nesse cenário Usuário do Sistema, Consumidor, Fornecedor e Endereço representam uma agregação. Não foi encontrado objeto de valor nesse domínio e foi definido um serviço chamado Usuário para controlar o registro dos dados do usuário e o controle de acesso.

De posse do resultado do DDD tático para cada contexto de domínio limitado, foi iniciada a fase de designação dos microserviços. Para isso, seguindo os passos descritos em Microsoft (2019c):

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, acabavam mapeando e correlacionando as entidades totais do domínio onde se encontravam, portanto, dessa análise não resultou microserviços.
3. Cada serviço de domínio foi eleito como microserviço. Assim foram elencados os microserviços: Gerenciamento do Usuário, Repasse ao Fornecedor, Avaliação, Suporte, Entrega, Pedido, Pagamento, Carrinho, Catálogo e Estoque.
4. Cada microserviço citado foi analisado e em função de Gerenciamento de Usuário possuir mais que uma responsabilidade que poderia contemplar as características dos microserviços, surgiu o último microserviço definido para essa arquitetura: Controle de Acesso ao Usuário.

Na Figura 8 é mostrada parte da arquitetura de microserviços definida neste trabalho para uma aplicação de marketplace. Considerando que cada microserviço pode ser implementado como se fosse uma aplicação independente, cada um dos microserviços pode ter tecnologia completamente distinta a ser empregada e pode evoluir independente de outros microserviços. Assim, fica clara uma das grandes vantagens, como a manutenção e escalabilidade, desse tipo de arquitetura frente à arquitetura monolítica. Entretanto, como pode ser observado, como cada microserviço tem sua própria infraestrutura, inclusive de banco de dados, então a alteração de uma entidade como Produto em um microserviço pode acarretar a comunicação para que outro microserviço atualize parte dos seus dados para essa entidade.

Figura 8 – Arquitetura de microserviços para marketplace.



Fonte: Próprio Autor.

5- Considerações Finais

Durante o desenvolvimento do presente trabalho, houve uma dificuldade para identificar os microserviços do e-commerce, e isso ocorreu devido a maioria dos artigos e trabalhos de referência não se aprofundarem em metodologias para defini-los.

Em suma, foi necessário encontrar um padrão de modelagem que fornecesse uma estrutura como base para a definição dos microserviços do sistema, e não apenas discutisse a importância de desenvolver os mesmo de forma precisa a fim de evitar problemas.

Assim, ao escolher empregar o padrão de modelagem de projeto Domain Driven Design (DDD) este trabalho cumpre o objetivo de mostrar os passos que podem ser seguidos, segmentando o processo de desenvolvimento do marketplace desde a identificação do contexto do sistema até definição dos microserviços.

Com relação a escalabilidade do sistema, é importante ressaltar que uma arquitetura de microserviços no contexto de e-commerce, se torna essencial uma vez que permite o gerenciamento e acréscimo de tecnologias em contextos de domínios independentes.

Porém, esse tipo de arquitetura acarreta maior complexidade e um maior nível de planejamento conforme o sistema cresce, desta forma não se torna ideal para softwares de pequeno porte ou mesmo com funcionalidades simples.

Por isso, é necessário um planejamento inicial com os stakeholders a fim de definir o tamanho do sistema, para assim escolher a abordagem que melhor atende ao cenário em questão, evitando despesas futuras ou um aumento na complexidade da aplicação.

Como trabalhos futuros, a arquitetura aqui definida pode ser analisada e incrementada frente aos requisitos não funcionais da aplicação. Além disso, aspectos de infraestrutura podem ser acrescentados para que ela possa se tornar uma arquitetura de referência ou mesmo passível de ser prototipada.

Referências

AMARAL, José Diogo Coelho. **A evolução das arquiteturas monolíticas para as arquiteturas baseadas em microserviços.** 2018. Disponível em: <https://recipp.ipp.pt/bitstream/10400.22/11920/1/DM_JoseAmaral_2018_MEI.pdf>. Acesso em: 08/06/2020.

AULKEMEIER, Fabian; SCHRAMM, Milan; LACOB, Maria-Eugenia; HILLEGERSBERG, Jos van. **A service-oriented e-commerce reference architecture.** 2016. Disponível em: <https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-18762016000100003>. Acesso em: 20/03/2020.

AULKEMEIER, Fabian; ANGGASTA, Mohammad; LACOB, Maria-Eugenia; HILLEGERSBERG, Jos van. **A pluggable service platform architecture for e-commerce.** 2016. Disponível em: <<https://link.springer.com/article/10.1007/s10257-015-0291-6>>. Acesso em: 20/03/2020.

AZEVEDO, Ana Rita Martins. **A adoção do e-commerce e a velocidade de internacionalização das empresas.** 2015. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/80763/2/36742.pdf>>. Acesso em: 06/05/2020.

BAGATINI, Laimer. **O contexto do e-commerce no brasil: análise do desempenho do varejo online no período de 2003 a 2018.** 2019. Disponível em: <<http://bibliotecadigital.fgv.br/ocs/index.php/clav/clav2019/paper/view/7155/2041>>. Acesso em: 06/05/2020.

CHINAGLIA, Rafael. **E-commerce cresce 12% no primeiro semestre de 2019, segundo 40º webshoppers.** Redação E-Commerce Brasil, 2019. Disponível em: <<https://www.ecommercebrasil.com.br/noticias/e-commerce-cresce-12-por-cento-webshoppers-i-e-commerce-brasil/3/>>. Acesso em: 29/03/2020.

CORRÊA, Andreiuid Sheffer. **Uma arquitetura de referência colaborativa para estruturação de dados abertos governamentais**. 2017. Disponível em: <<https://teses.usp.br/teses/disponiveis/3/3141/tde-28062017-101825/publico/AndreiuidShefferCorreaCorr17.pdf>>. Acesso em: 05/05/2019.

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

HAIDER, U.; WOODS, E.; BASHROUSH, R. **Representing variability in software architecture: a systematic literature review**. 2018. Disponível em: <<https://repository.uel.ac.uk/item/8474q>>. Acesso em: 03/05/2020.

JÚNIOR, Elemar. **Seis benefícios da arquitetura de software (bem-feita!)**. 2018. Disponível em: <<https://www.eleamarjr.com/pt/2018/03/seis-beneficios-da-arquitetura-de-software-bem-feita/>>. Acesso em: 29/03/2020.

MALDONADO, José Carlos; BRAGA, Rosana Teresinha Vaccare; GERMANO, Fernão Stella Rodrigues; MASIERO, Paulo Cesar. **Padrões e frameworks de software (notas didáticas)**. USP, 2002. Disponível em: <<https://sites.icmc.usp.br/rtvb/apostila.pdf>>. Acesso em: 08/05/2020.

MAGALHÃES, José Nuno Freitas. **Definição dos períodos de picking num negócio de e-commerce**. 2015. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/89066/2/34456.pdf>>. Acesso em: 05/05/2020.

MENDONÇA, Herbert Garcia de. **E-commerce**. 2016. Disponível em: <<http://www6.uninove.br/ojs/journaliji/index.php/iptec/article/view/68/146>>. Acesso em: 05/05/2020.

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: 03/05/2020.

MICROSOFT. **Usando a análise de domínio para modelar microsserviços**. 2019a. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/model/domain-analysis>>. Acesso em: 03/05/2020.

MICROSOFT. **Usando DDD tático para projetar microsserviços**. 2019b. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/model/tactical-ddd>>. Acesso em: 03/05/2020.

MICROSOFT. **Identificando limites de microsserviço**. 2019c. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/microservices/model/microservice-boundaries>>. Acesso em: 03/05/2020.

RICHARDSON, Chris. **Pattern: microservice architecture**. 2020. Disponível em: <<https://microservices.io/patterns/microservices.html>>. Acesso em: 07/05/2020.

SBVC – SOCIEDADE BRASILEIRA DE VAREJO e CONSUMO. **Ranking das 50 maiores empresas do e-commerce brasileiro 2018 – um estudo completo do e-commerce, 2018**. 2018.

SOUZA, Ítalo Andrade de; PELISSARI, William Roberto. **Microsserviços como alternativa de arquitetura monolítica**. 2017. Disponível em: <<https://www.revista.unifcv.edu.br/index.php/revistapos/article/view/66/79>>. Acesso em: 08/06/2020.

TAKAHATA, Leonardo Eidy. **E-commerce no marketplace**. 2017. Disponível em: <<https://www.riuni.unisul.br/handle/12345/4897>>. Acesso em: 08/05/2020.

TARAS, André Krzyk; VICHINHESKI, Bruno. **Modelagem de uma ferramenta para geração automática de e-commerce B2B e B2C usando DDD**. 2016. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/7368>>. Acesso em: 11/06/2020.

VIDIGAL, Arthur Rocha; BENFEITA, Polyana Rocha. **Inovações no varejo brasileiro: o caso Magazine Luiza**. Projeto de Graduação apresentado ao curso de Engenharia de Produção da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenharia de Produção. 2019. Disponível em: <<http://monografias.poli.ufrj.br/monografias/monopoli10030671.pdf>>. Acesso em: 02/08/2020.

WU, Xinyi; GEREFFI, Gary. **Amazon and Alibaba: Internet governance, business models, and internationalization strategies**. International Business in the Information and Digital Age Progress in International Business Research, Volume 13, 327-356, 2019. Publicado em: <<https://www.emerald.com/insight/content/doi/10.1108/S1745-886220180000013014/full/pdf?title=italicamazon-and-alibabaitalic-internet-governance-business-models-and-internationalization-strategies>>. Acesso em: 02/08/2020.