

Aplicação do Desenvolvimento Baseado em Domínio (DDD) na Criação de uma Ferramenta para Geração Automática de E-Commerce B2B E B2C

André Krzyk Taras
Departamento Acadêmico de
Informática
Universidade Tecnológica Federal
do Paraná
Ponta Grossa - Paraná
andrektaras@gmail.com

Bruno Vichinheski
Departamento Acadêmico de
Informática
Universidade Tecnológica Federal
do Paraná
Ponta Grossa - Paraná
brunovichinheski@gmail.com

Simone Nasser Matos
Departamento Acadêmico de
Informática
Universidade Tecnológica Federal
do Paraná
Ponta Grossa - Paraná
snasser@utfpr.edu.br

RESUMO

O comércio eletrônico (*e-commerce*) é utilizado por milhões de pessoas e pode ser dividido em categorias tais como: empresa para empresa (B2B), consumidores (B2C), governos (G2G), dentre outras. Existem diversas plataformas para criação de *e-commerce*, porém atendem apenas uma categoria. Este trabalho aplicou o *Domain-Drive Design* (DDD) para a criação dos módulos fundamentais que compõem uma plataforma *e-commerce* nas categorias B2B (*Business To Business*) e B2C (*Business To Customers*). O padrão de desenvolvimento DDD foi escolhido pela: necessidade de compreensão do domínio, alta manutenibilidade e capacidade de solucionar problemas intrínsecos nos sistemas de *e-commerce* atuais.

Palavras-Chave

B2B. B2C. DDD.

ABSTRACT

E-commerce is used by millions of people and can be divided by categories like Business To Business (B2B), Business To Customers (B2C) or Government To Government (G2G), among others. There are platforms used to create e-commerce, however it's only destined to one category. This paper applied Domain-Drive Design (DDD) to create fundamental modules that compose an e-commerce platform in B2C and B2B categories. The DDD pattern was chose to understand the complexity problem, maintainability and ability to solve specific problems in the current e-commerce.

Keywords

B2B. B2C. DDD.

1. INTRODUÇÃO

Uma plataforma e-commerce é responsável por realizar

transações de compra e venda de produtos e serviços por meio da Internet. Essa plataforma possibilita a interação entre compradores e vendedores de diferentes distâncias [1].

O sistema e-commerce possui um alto crescimento a cada ano; no Brasil o faturamento desse comércio eletrônico chegou a R\$48,5 bilhões em 2015 e para 2016 a estimativa é que alcance R\$56,8 bilhões [3].

A evolução da tecnologia e do comércio eletrônico, fez com que muitas empresas desenvolvessem plataformas e-commerce para seus clientes aumentarem a competitividade de venda de seus produtos. Tais plataformas possuem alto nível de complexidade, porém, muitas são desenvolvidas sem grande planejamento, impedindo o seu desenvolvimento, deixando os seus clientes com um sistema engessado e que não acompanha as tendências de mercado [2].

A utilização de plataformas e-commerce propicia a necessidade de um produto com baixo custo de manutenção, de modo que possa acompanhar as constantes novidades das tecnologias empregadas no seu desenvolvimento de forma eficaz e com isso proporcione um preço compatível.

Diversas metodologias de desenvolvimento podem ser utilizadas quando se deseja construir um sistema de forma padronizada e planejada. O modelo de desenvolvimento proposto por Erick Evans [4], *Domain-Drive Design* (DDD), foi o escolhido para o desenvolvimento deste trabalho pois possui como objetivo construir um software de fácil manutenção e é indicado para projetos que contemplam domínios complexos assim como o de e-commerce.

Evans [4] afirma que utilizar DDD é um desafio, pois requer um grande esforço da equipe de desenvolvimento e um conhecimento aprofundado dos objetivos deste software, porém ressalta vantagens de possuir um sistema de fácil manutenção e a capacidade de não se tornar obsoleto quando novas tecnologias surgirem devido a sua capacidade de adaptação.

Este trabalho aplicou o DDD na criação de uma plataforma capaz de gerar e-commerce B2B e B2C, utilizando o framework *Symfony* e algumas bibliotecas para a camada de infra-estrutura, proporcionando flexibilidade à aplicação.

2. APLICAÇÃO DO DDD

Para a criação da plataforma a solução foi aplicar o desenvolvimento baseado em domínio (DDD) que proporciona gerar um modelo adaptável e flexível [4].

O modelo de desenvolvimento DDD é composto pelas melhores práticas de desenvolvimento e os padrões de projeto mais

utilizados, sua abordagem de desenvolvimento divide-se em quatro etapas [4]:

- Trabalhar o modelo do domínio;
- Construir o modelo baseado em domínio;
- Refatorar o modelo;
- *Design* estratégico;

Ao adotar a estratégia de desenvolvimento baseado em domínio, a implementação irá se espelhar no modelo, portanto, a maioria dos esforços será em seu aperfeiçoamento.

2.1 Trabalhar o modelo do domínio

Nesta etapa do desenvolvimento foi realizado um estudo com o propósito de analisar todos os módulos e funcionalidades presentes nas categorias B2C e B2B, assim como mostra a Tabela 1. Após este levantamento foram definidos dois módulos para implementação, os módulos escolhidos foram "Cadastro e Login de Usuários" e "Catálogo de Produtos".

Tabela 1. Módulos e funcionalidades das Categorias

| Módulo | Funcionalidade | Categoria |
|--|---|--|
| Cadastro e Login de Usuários | <ul style="list-style-type: none"> ▪ Incluir Usuário ▪ Definir Tipo de Usuário ▪ Realizar Login | <ul style="list-style-type: none"> ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C |
| Catálogo de Produtos | <ul style="list-style-type: none"> ▪ Incluir Produto ▪ Listar Produtos | <ul style="list-style-type: none"> ▪ B2B e B2C ▪ B2B e B2C |
| Relacionamento entre Cliente e Empresa | <ul style="list-style-type: none"> ▪ Contrato ▪ Atendimento ao Cliente ▪ Suporte ao Cliente | <ul style="list-style-type: none"> ▪ B2B ▪ B2B e B2C ▪ B2B e B2C |
| Transação e Entrega | <ul style="list-style-type: none"> ▪ Adicionar Produto ▪ Excluir Produto ▪ Alterar Quantidade do Produto ▪ Calcular o Frete ▪ Calcular o Valor do Produto ▪ Verificar Formas de Pagamento ▪ Verificar Cupom Desconto ▪ Confirmar Pagamento ▪ Finalizar Compra ▪ Cancelar Compra ▪ Verificar Carrinho ▪ Verificar Login ▪ Verificar Endereço para Entrega | <ul style="list-style-type: none"> ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C ▪ B2B e B2C |

Esses módulos foram escolhidos e implementados com o objetivo de mostrar que ambos possuem uma funcionalidade que é exatamente igual (Cadastro e Login de Usuários) e outra que apresenta uma pequena variação (Catálogo de Produtos), pois na categoria B2B pode-se aplicar regras para mostrar os produtos aos usuários, por exemplo, se o usuário tiver efetuado o login no website, este possui alguns benefícios ao visualizar os produtos podendo exibir por: preços, quantidade de itens disponíveis, entre outros detalhes [6]. Na plataforma B2C a relação de produtos mostrados aos usuários é a mesma tanto para um que tenha

efetuado o login quanto para outro que apenas está navegando no website.

2.2 Construir o modelo baseado em domínio

Para a construção do modelo de e-commerce foi necessário conhecer as principais funcionalidades das categorias B2B e B2C e isolá-las de forma a obter um modelo comum para ambas. Com isso o modelo e-commerce foi dividido em camadas conforme descreve o DDD: interface, aplicação, domínio e infra-estrutura. A camada de infra-estrutura é a primeira relatada devido a utilização de um framework para organização e controle de toda a estrutura do projeto.

2.2.1 Camada de infra-estrutura

Para o início da implementação optou-se pela escolha de um framework que pudesse atender os requisitos desta camada. Foram realizadas pesquisas e testes com os frameworks mais conhecidos para aplicações WEB (Zend, CakePHP, Symfony) e optou-se pela utilização do Symfony 3 [5], pois foi o framework com a melhor documentação e recursos disponíveis dentre os testados.

Na camada de infraestrutura o framework Symfony objetiva realizar a organização do trabalho, bem como a interação e persistência dos dados entre as camadas. Para o processo de instalação do framework Symfony foi necessário fazer o download do código-fonte [5]. Depois disso, foi realizado o download do Composer (ferramenta que gerencia dependências em PHP e permite a declaração de bibliotecas no projeto, bem como a instalação e atualização das mesmas).

2.2.2 Camada de interface

A camada de interface do sistema e-commerce interage diretamente com o usuário através de formulários, listagens de produtos, entre outros. O framework Symfony possui um diretório padrão para todas as interfaces. As interfaces para os módulos de cadastro e login de usuários e catálogo de produtos foram implementadas em HTML e TWIG, este último compila templates em código PHP otimizado, isto faz com que os códigos se tornem mais rápidos, seguros e flexíveis.

2.2.3 Camada de aplicação

Uma importante ferramenta disponível no framework Symfony são os Controllers, responsáveis por receber requisições HTTP, criar e retornar uma resposta HTTP.

Uma pasta Controller foi criada para reunir os controladores necessários na aplicação, a função de um controlador é renderizar o conteúdo de uma página ou até mesmo redirecionar para uma página de erro caso a página não exista. Os controladores dos módulos de Cadastro e Login de Usuários e Catálogo de Produtos têm como objetivo gerenciar uma requisição específica de um usuário, como por exemplo, a confirmação de um formulário de cadastro ou a visualização da página de um produto.

2.2.4 Camada de domínio

Para modelagem do DDD devem ser seguidos sete blocos de construção que fundamentam e isolam o domínio do restante da aplicação: entidades, objetos de valores, serviços, módulos, agregados, fábricas e repositórios. Para o desenvolvimento dos módulos de Cadastro e Login de Usuários e Catálogo de Produtos foram utilizados apenas quatro blocos de construção: entidades, serviços, módulos e repositórios, não foram encontradas aplicações para os demais blocos nessa implementação.

O desenvolvimento da camada do domínio iniciou-se com a compreensão dos blocos de construção e a divisão correta dos elementos de uma aplicação e-commerce. Em uma pasta chamada

Entity são armazenadas as entidades do domínio que foram identificadas nos módulos implementados, cada classe contém uma assinatura (@Entity), que é identificada pelo Doctrine (biblioteca padrão do Symfony que gerencia o banco de dados) como uma tabela do banco de dados.

2.3 Refatorar o modelo

Durante o processo de refatoração nesse trabalho, não foram aplicadas técnicas de refatoração. O objetivo da refatoração foi obter um melhoramento dos modelos e aplicar padrões de projeto. A Tabela 2 mostra os módulos trabalhados (aqueles que foram identificados na primeira etapa do DDD), o respectivo número de refatorações e uma breve descrição sobre o que foi refatorado.

Tabela 2. Módulos Refatorados

| Módulo | Refatorações | Descrição |
|--|--------------|---|
| Cadastro e <i>Login</i> de Usuários | 2 | <ul style="list-style-type: none"> Exclusão de classes redundantes e tornando-as atributos Melhoramento do relacionamento entre as classes |
| Catálogo de Produtos | 3 | <ul style="list-style-type: none"> Incluso a classe <i>Repositorio</i> para realizar consultas Incluso as classes de busca Inclusão de métodos diferenciados na categoria B2B |
| Relacionamento entre Cliente e Empresa | 1 | <ul style="list-style-type: none"> Melhoramento da classe <i>Compra</i> para suprir ambas as categorias de <i>e-commerce</i> |
| Transação e Entrega | 5 | <ul style="list-style-type: none"> Inclusão da classe <i>ItemCarrinho</i> Inclusão da classe <i>Rota</i>; mudança dos atributos da classe <i>Transportadora</i> Inclusão da classe <i>Compra e Venda</i> Generalização das classes <i>Transportadora</i> e <i>Filial</i> como classe <i>PessoaJuridica</i> Inclusão das classes: <i>ProdutosCompradosLoja</i>, <i>Movimentação e Saldo</i> |

2.4 Design Estratégico

As fases iniciais do DDD são importantes para o desenvolvimento do produto final, porém a parte mais relevante desta estratégia é quando o sistema torna-se complexo, por isso se deve dividir o sistema em contextos delimitados. Para isto, utiliza-se técnicas para a manipulação e compreensão de grandes modelos [4].

Como esta etapa possui um foco no final do desenvolvimento e também quando existe algum tipo de integração com outros sistemas, ela não foi utilizada nesta pesquisa.

3. RESULTADOS

A página inicial desenvolvida para a ferramenta apresenta as opções de e-commerce disponíveis pela aplicação, ou seja, as categorias B2C e B2B, assim como mostra a Figura 1.



Figura 1. Página inicial da ferramenta

A Figura 1 apresenta apenas o protótipo do que seria o produto final, pois este se encontra em fase de desenvolvimento, apenas a modelagem desta aplicação base foi realizada.

Após a escolha da categoria de e-commerce o usuário escolhe quais módulos a plataforma terá, os módulos fundamentais que já estão implementados obrigatoriamente deverão fazer parte da aplicação, desta forma este modelo contempla módulos extras que poderão ser desenvolvidos e optativamente utilizados pelos usuários. A etapa de isolamento do domínio exigiu grande conhecimento do sistema e demandou grandes esforços dos envolvidos para obter o conhecimento necessário para o desenvolvimento, entretanto, este processo tornou o modelo mais robusto, claro e flexível, possibilitando a adição de funcionalidades extras sem que muitas alterações sejam necessárias.

Como foram encontradas diferenças para o catálogo de produtos entre as categorias B2C e B2B, foram criadas duas implementações diferentes para cada categoria e-commerce. Para o catálogo de produtos B2C foi implementado um catálogo em que o usuário pode visualizar os produtos com informações de preço disponíveis, independente se ele é um usuário que está logado no sistema ou navega anonimamente, como mostra a Figura 2.

Na implementação do catálogo B2B, um usuário tem acesso limitado as informações dos produtos e precisa fazer parte de um grupo de usuários específico para ter acesso aos preços e outras informações importantes, geralmente este grupo de usuários ganha este acesso através de concessões administrativas. A regra de negócio irá depender da empresa que vai gerenciar o e-commerce.

Para o catálogo B2B desta aplicação é verificado se o usuário está logado e se ele pertence ao grupo privilegiado, caso contrário ele não terá acesso as informações referentes ao preço do produto.

A Figura 2 ilustra as diferenças encontradas no catálogo de produtos das categorias durante a listagem dos resultados de uma busca.

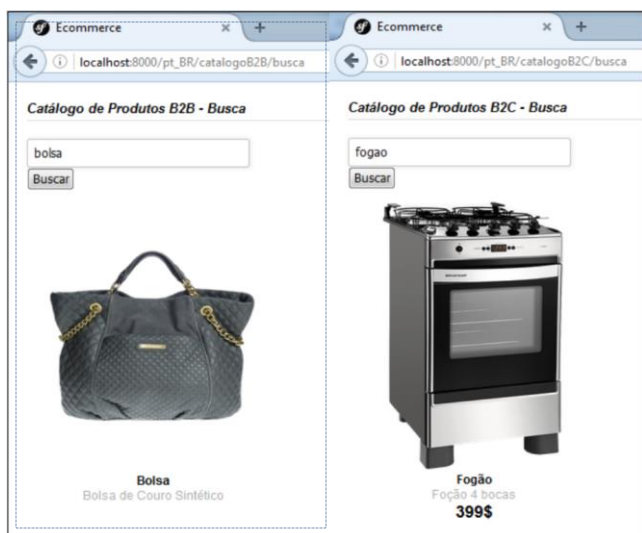


Figura 2. Diferenças do catálogo B2B e B2C na Busca de Produtos

É possível notar que a busca B2B (destacada na imagem) não possui nenhuma informação referente ao preço do produto, enquanto que o catálogo mostrado para a busca B2C apresenta o preço normalmente.

4. CONCLUSÃO

A modelagem e implementação dos módulos login e catálogo de produtos nas categorias B2B e B2C foram realizadas seguindo a metodologia de desenvolvimento baseado em domínio. Primeiramente foram realizados modelos iniciais, em seguida utilizou-se os blocos de construção para isolar o domínio do restante da aplicação, a partir deste ponto prevaleceram as refatorações. A etapa de *design* estratégico do DDD não foi utilizada.

O modelo da ferramenta proposta difere das plataformas de geração de *e-commerce* porque contempla diversas categorias, diferente das outras que atendem apenas uma. O desenvolvimento baseado em domínio proporcionou flexibilidade ao modelo

porque funcionalidades podem ser acrescentadas sem que haja grandes mudanças na estrutura do modelo.

Embora a implementação do modelo não esteja completa, pôde-se constatar que a utilização do DDD para a criação do modelo inicial foi fundamental, a utilização das etapas e dos blocos de construção possibilitaram um entendimento graduado do domínio do sistema pois focou-se em compreender as funcionalidades do domínio, minimizando os esforços para o desenvolvimento de interfaces e funções menos importantes do sistema. O uso do framework foi importante para o processo de desenvolvimento, toda a camada de infra-estrutura do DDD foi suprida pelo *Symfony*.

5. REFERÊNCIAS

- [1] ASCENSÃO, Carlos P. O que é e-Commerce?. 2014. Disponível em: <<http://www.gestordeconteudos.com/tabid/3850/Default.aspx>>. Acesso em: 24 nov. 2015.
- [2] CHAUSSARD, Cristiano. Três erros cometidos por plataformas de e-commerce. 2015. Disponível em: <<http://www.guiadeecommerce.com.br/tres-erros-cometidos-por-plataformas-de-ecommerce/>>. Acesso em: 09 set. 2016.
- [3] DOTSTORE (Mogi das Cruzes). E-Commerce 2016: Projeção de R\$56,8 bilhões em faturamento. 2016. Disponível em: <<http://site.dotstore.com.br/loja-virtual/e-commerce-2016-projecao-de-r568-bilhoes-em-faturamento/>>. Acesso em: 12 maio 2016.
- [4] EVANS, Eric. Domain-Drive Design: Tackling Complexity in the Heart of Software. Prentice Hall, 2003.
- [5] FABIEN POTENCIER (California) (Org.). SYMFONY. Página do framework. Disponível em: <<http://symfony.com/>>. Acesso em: 15 mar. 2016.
- [6] SILVA, Edson dos Santos. Branding para empresas B2B. Fundação Instituto de Administração, Pós Graduação em Gestão Estratégica de Marcas. São Paulo: Provar, 2013. Disponível em: <http://www.cidademarketing.com.br/2009/sysfotos/tesesmo-no/tesem_40b478b412cce7278d88fb18869ca034.pdf>. Acesso em: 17 out. 2015.