

Utilizando processos de desenvolvimento de software para modelar e implementar um sistema de cadastros

Bruno Borges Luza, Jonatan Amaral da Silva e Marcos Klóss

Junho 2022

1 Introdução

Uma importante instituição de pesquisa, ensino e inovação - a TecnoUCS - integrada à Universidade de Caxias do Sul (UCS), manifestou a necessidade da implementação de uma plataforma que centralizasse os dados e informações de todo o conhecimento gerado pela instituição. O conhecimento gerado, que são os projetos, varia desde projetos disciplinares e multidisciplinares, até TCC, mestrados, doutorados, MBA e pesquisas científicas. Cada projeto gera um resultado e o conjunto de resultados também deve ser centralizado no sistema. O sistema também deve ser capaz de cadastrar os pesquisadores e relacioná-los com os projetos, que por sua vez estarão relacionados com os resultados. O objetivo desta centralização é facilitar o acesso às informações dos projetos, para que os pesquisadores não dupliquem seus esforços.

Neste contexto, foi possível aplicar uma espécie de mistura de processos de desenvolvimento de software para modelar e implementar o sistema. Os processos utilizados incluem Scrum, BDD (*Behavior Driven Development*), DDD (*Domain-Driven Design*) e TDD (*Test Driven Development*). O objetivo deste artigo é apresentar as etapas da modelagem e implementação do sistema utilizando os processos.

Segundo Pressman, o Scrum é um método de desenvolvimento ágil de software, que possui princípios coerentes com o manifesto ágil e que incorpora as seguintes atividades metodológicas: requisitos, análise, projeto, evolução e entrega [1]. As tarefas de cada atividade metodológica são realizadas no período de tempo chamado *sprint*. No projeto, o Scrum constituiu a base do processo de desenvolvimento, sugerindo a ordem das atividades. O processo foi dividido primeiramente em duas etapas: a etapa de modelagem e a etapa de implementação. Na etapa de modelagem, foram realizadas as atividades de requisitos e análise. Enquanto isso, na etapa de implementação, foram realizadas as atividades de projeto, evolução e entrega.

O BDD é um conjunto de práticas de engenharia de software projetadas para ajudar os times de desenvolvimento a desenvolver e entregar software de

alta qualidade [2]. Ferguson afirma que o BDD não é um substituto para o Scrum, XP ou outros métodos, mas sim que ele incorpora, desenvolve e aprimora ideias destas metodologias [3]. Por este motivo, o BDD pode ser inserido no processo com o objetivo de aprimorar uma de suas atividades. A atuação do BDD foi inserida na tarefa de formulação das histórias. Uma peculiaridade do BDD, e por isso ele é vantajoso ao ser utilizado para a formulação das histórias, é que o time de desenvolvimento, os usuários e outros *stakeholders* trabalham juntos para definir cenários que atendem às suas expectativas [4]. Além disso, a história é formulada em uma estrutura pronta, que utiliza as expressões *Given...When...Then*. Na etapa de modelagem será possível ver esta estrutura.

Em 2003, Eric Evans definia o que era DDD: uma abordagem de desenvolvimento de software que permite que os times de desenvolvimento possam construir e fazer a manutenção em projetos com domínios de negócio complexos [5]. Alguns princípios definidos pelo DDD são muito importantes para o desenvolvimento de software. Scott Millett descreve os seguintes princípios: focar no domínio principal, aprender através da colaboração entre os membros da equipe, criar modelos por meio de exploração e experimentação, criar uma linguagem única para melhorar a comunicação da equipe, compreender a aplicabilidade de um modelo e evoluir continuamente o modelo [6]. O DDD pode ser utilizado de forma bastante proveitosa para definir e descrever as entidades de domínio. Seus princípios foram aplicados no capítulo 2, onde será demonstrada sua utilização para descrever as entidades de domínio.

2 Modelagem do sistema

A primeira etapa na modelagem do sistema foi realizar o levantamento de requisitos para construir o backlog (tabela 1).

História	Título
1	Cadastrar pesquisador
2	Cadastrar resultado
3	Cadastrar projeto
4	Listar pesquisadores cadastrados
5	Listar projetos cadastrados

Tabela 1: backlog

O cadastro de pesquisador (1) permite que os pesquisadores - docentes, alunos e funcionários - possam ser cadastrados com o objetivo de serem relacionados posteriormente no cadastro de projeto. O cadastro de resultados obtidos (2) foi uma necessidade importante destacada pela instituição de pesquisa já que, a partir desta possibilidade, os pesquisadores poderiam acessar o projeto desejado e ter acesso ao resultado obtido de forma rápida e prática. Por este motivo, foi o segundo requisito considerado, e que será essencial para o cadastro do projeto. Cadastrar um projeto (3) é o pilar central do sistema. Nele é possível relacionar os pesquisadores com o projeto e o projeto com os resultados obtidos, relacionando as três entidades principais.

Os últimos dois requisitos, a listagem de pesquisadores cadastrados (4) e a listagem de projetos cadastrados (5), permite que o usuário possa encontrar facilmente a informação desejada a partir de tabelas com design intuitivo. Cumpre a necessidade do acesso prático à informação, destacado pela instituição.

As histórias foram descritas utilizando a abordagem do BDD. Elas são listadas nas tabelas 2 a 6, correspondentes às histórias do backlog apresentado na tabela 1.

Título: cadastrar pesquisador
Cenário: como pesquisador, eu gostaria de realizar meu cadastro
Dado que um pesquisador deseja criar um cadastro e o pesquisador não esteja cadastrado no sistema Quando as informações pessoais forem preenchidas e a área de graduação principal estiver preenchida e as áreas e subáreas estiverem informadas Então garanta que o pesquisador seja cadastrado.

Tabela 2: cadastrar pesquisador

Título: cadastrar resultado
Cenário: como pesquisador, eu gostaria de cadastrar o resultado do meu projeto
Dado que um projeto esteja cadastrado Quando um, ou mais, arquivos forem selecionados para upload Então garanta que o resultado seja cadastrado e que projeto e resultado sejam vinculados em banco de dados.

Tabela 3: cadastrar resultado

Título: cadastrar projeto
Cenário: como pesquisador, eu gostaria de cadastrar meu projeto
Dado que o projeto não tenha sido cadastrado e o pesquisador já tenha sido cadastrado Quando o usuário digitar o título do projeto e selecionar a área de conhecimento e selecionar os pesquisadores, ordenados por afinidade com a área e selecionar, ou não, o tipo de financiamento e selecionar, ou não, a opção de finalizado e inserir, ou não, a data de término do projeto e cadastrar, ou não, o resultado gerado Então garanta que o projeto seja cadastrado e que projeto, pesquisador e resultado sejam vinculados em banco de dados.

Tabela 4: cadastrar projeto

Título: listar pesquisadores cadastrados
Cenário: como um administrador do sistema, eu gostaria de visualizar todos os pesquisadores já cadastrados
Dado que pesquisadores estejam cadastrados Quando o administrador do sistema acessar a tela de pesquisadores Então garanta que os pesquisadores sejam listados e possam ser filtrados.

Tabela 5: listar pesquisadores cadastrados

Título: listar projetos cadastrados
Cenário: como pesquisador, eu gostaria de visualizar todos os projetos já cadastrados
Dado que projetos estejam cadastrados Quando o pesquisador acessar a tela de projetos Então garanta que os projetos sejam listados em uma tabela e possam ser filtrados.

Tabela 6: listar projetos cadastrados

Escrever utilizando a estrutura sugerida pelo BDD possibilitou que as histórias fossem descritas de maneira didática, evidenciando ainda mais os requisitos. Além disso, a equipe pôde concluir que, utilizando essa estrutura, houve praticidade e agilidade na formulação da história, auxiliando também no processo de esmiuçar os requisitos.

3 Referência bibliográficas

- [1] Roger S. Pressman, Bruce R. Maxim. Engenharia de Software, 9^a edição, pp. 41, 15 de junho de 2011.
- [2] John Ferguson Smart. BDD in Action, 1^a edição, pp. 12, 12 de outubro de 2014.
- [3] John Ferguson Smart. BDD in Action, 1^a edição, pp. 4, 12 de outubro de 2014.
- [4] John Ferguson Smart. BDD in Action, 1^a edição, pp. 18, 12 de outubro de 2014.
- [5] Scott Millett, Nick Tune. Patterns, Principles and Practices of Domain-Driven Design. Edição ilustrada, pp. 3, 4 de maio de 2015.