

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI
Programa de Pós-Graduação em Tecnologia, Ambiente e Sociedade
Yvssa Carneiro Desmots Eliote

**IMPLANTAÇÃO E ANÁLISE DO *FRAMEWORK SCRUM* NO
DESENVOLVIMENTO DA PLATAFORMA ABERTA NOSSO EXERCÍCIO**

Teófilo Otoni – MG – Brasil
2018

Yvssa Carneiro Desmots Eliote

**IMPLANTAÇÃO E ANÁLISE DO *FRAMEWORK SCRUM* NO
DESENVOLVIMENTO DA PLATAFORMA ABERTA NOSSO EXERCÍCIO**

Dissertação apresentada ao Programa de Pós-Graduação *Stricto Sensu* Mestrado em Tecnologia, Ambiente e Sociedade (PPGTAS) da Universidade Federal dos Vales do Jequitinhonha e Mucuri como requisito parcial para obtenção do título de mestre.

Orientador: Prof. Dr. Wederson Marcos Alves
Coorientador: Prof. Dr. Mauro Lúcio Franco.

Teófilo Otoni – MG - Brasil

2018

FICHA CATALOGRÁFICA

Ficha Catalográfica
Preparada pelo Serviço de Biblioteca/UFVJM
Bibliotecário responsável: Gilson Rodrigues Horta – CRB6 nº 3104

E429i Eliote, Yvssa Carneiro Desmots.
2018 Implantação e análise do framework Scrum no desenvolvimento da plataforma aberta Nosso Exercício./Yvssa Carneiro Desmots Eliote. Teófilo Otoni, 2018.
113p. ; il.

Dissertação (Mestrado) – Universidade Federal dos Vales do Jequitinhonha e Mucuri. Programa de Pós-Graduação em Tecnologia, Ambiente e Sociedade, 2018.

Orientador: Prof. Dr. Wederson Marcos Alves
Coorientador: Prof. Dr. Mauro Lúcio Franco.

1. Engenharia de software. 2. Metodologia ágil. 3. Framework Scrum. 4. Desenvolvimento de software. 5. Pesquisa-ação. I. Título.

CDD: 004

Yvssa Carneiro Desmtos Eliote

**IMPLANTAÇÃO E ANÁLISE DO *FRAMEWORK SCRUM* NO
DESENVOLVIMENTO DA PLATAFORMA ABERTA NOSSO EXERCÍCIO**

Dissertação apresentada ao Programa de Pós-Graduação Stricto Sensu Mestrado Profissional em Tecnologia, Ambiente e Sociedade (PPGTAS) da Universidade Federal dos Vales do Jequitinhonha e Mucuri como requisito parcial para obtenção do título de mestre.

Orientador: Prof. Dr. Wederson Marcos Alves.

Data de aprovação ____/____/_____

Profa. Dra. Maria Lúcia Bento Villela
Universidade Federal dos Vales do Jequitinhonha e Mucuri

Prof. Dr. Marcio Coutinho de Souza
Universidade Federal dos Vales do Jequitinhonha e Mucuri

Prof. Dr. Wederson Marcos Alves
Universidade Federal dos Vales do Jequitinhonha e Mucuri

Teófilo Otoni – MG – Brasil

2018

Aos meus filhos amados, Lucas e Tiago, por serem a motivação que preciso para buscar ser
uma pessoa melhor a cada dia.

AGRADECIMENTOS

A Deus pela oportunidade que me concedeu de percorrer este caminho com saúde, dedicação, persistência e força.

Aos meus pais, Yvo e Clarice, pelos bons exemplos e incentivo. À Clara, minha irmã, que sempre acreditou em mim, que não mediu esforços para me amparar quando precisei, que me inspirou com sua dedicação e disciplina. Ao meu esposo, Dil, pela compreensão nos momentos de ausência e por acreditar nos meus sonhos. Aos demais familiares pelo apoio, carinho e compreensão.

A Cássio e Óseas pela parceria, amizade, risadas, conhecimentos compartilhados, vocês foram anjos que Deus enviou para aliviar esta trajetória.

A Ivana Carneiro, pelo impulso, bons conselhos e motivação. A Ahminne pela prestatividade e consideração.

Aos mestres, pela competência e comprometimento com os seus ensinamentos, em especial Márcio Coutinho e Mauro Lúcio Franco.

Ao Aquino pelo suporte, empenho e confiança neste trabalho.

A Luiz Fernando, pela colaboração, dedicação, superação e paciência.

Ao meu orientador, Wederson, pela postura ética, humildade, respeito e principalmente, pela sabedoria na condução do seu papel. Levarei seus ensinamentos e exemplo sempre comigo.

Sou imensamente grata a todos!

RESUMO

Este estudo propõe a implementação e análise do *Framework Scrum* no desenvolvimento de novas funcionalidade para o *website* Nosso Exercício. Esta aplicação *web* consiste em um dos projetos do Programa de Educação Tutorial, PET-UFVJM/Campus do Mucuri situada na cidade de Teófilo Otoni-MG e tem como finalidade o compartilhamento aberto de exercícios didáticos de diversas áreas do conhecimento. Por solicitação de seus idealizadores, novas funcionalidades foram demandadas para o Nosso Exercício, porém, não existiam para este projeto metas claras nem um plano de trabalho a ser seguido, os requisitos não estavam formalizados e a equipe disponível não estava madura nas tecnologias utilizadas. Levando-se em consideração o desafio enfrentado na geração de *software* de qualidade e o limite de tempo disponível para a realização desta pesquisa, foi proposto o uso de um processo da Engenharia de *Software* com intuito de se obter maior controle e qualidade do produto final a ser desenvolvido. O método ágil *Scrum* foi o escolhido para gerenciar as atividades de desenvolvimento para este *software*. Assim, o objetivo geral desta pesquisa consistiu em buscar uma resposta para a seguinte pergunta-problema: Quais benefícios e/ou dificuldades podem ser obtidas através da aplicação do *Framework Scrum* na evolução do desenvolvimento da plataforma aberta Nosso Exercício? Seguindo o modelo sugerido por Coughlan e Coughlan (2002), o método de pesquisa-ação foi utilizado para descrever a dinâmica conduzida durante este trabalho. A implantação do *Scrum* no Nosso Exercício ocorreu em duas etapas, a primeira, visou realizar uma capacitação sobre o *Scrum* e as ferramentas tecnológicas utilizadas no desenvolvimento do Nosso Exercício. Já a segunda, tratou do desenvolvimento das funcionalidades para este *website*. A análise dos resultados mostrou vários benefícios obtidos com a implantação do *Scrum* neste projeto, como: o foco e compromisso do Time durante o seu trabalho, o atendimento às reais necessidades do cliente (*Product Owner*), a flexibilidade do *framework* em se ajustar às condições de ambiente e trabalho de cada Time criado; o aprendizado contínuo do *software* e do processo resultante das discussões feitas nas Reuniões de Planejamento, Revisão e Retrospectiva. O ponto crítico do trabalho deu-se sobre a imaturidade com as tecnologias utilizadas para o desenvolvimento do *website*. As lições aprendidas com esta pesquisa indicam que é possível obter benefícios com a implantação do *framework Scrum* que superem as suas dificuldades, desde que sejam feitas as devidas análises do ambiente em que o mesmo for adotado.

Palavras chave: Engenharia de *Software*. Metodologia Ágil. *Framework Scrum*.

Desenvolvimento de *software*. Pesquisa-ação.

ABSTRACT

This study proposes an implementation and analysis of Scrum Framework in the development of new functionalities to 'Nosso Exercício' (Our Exercise) website. This web application consists in one of the Tutorial Education Program Projects, PET-UFVJM/Mucuri Campus, located in Teófilo Otoni city, Minas Gerais state and it has as its objective the open share of teaching exercises of several knowledge areas. At the request of its creators, new functionalities were demanded to 'Nosso Exercício' (Our Exercise), however, there were not clear goals for this project nor a work plan to be followed, the requirements were not formalised and the available team was not mature enough on the technologies applied. Considering the challenge faced on the generation of a good quality software and the time limit available for taking this research, the use of a software engineering was proposed aiming to get bigger quality control of the final Product to be developed. The agile method Scrum was the one chosen to manage the developing activities to this software. So, the general goal of this research consisted in searching for an answer to the following question-problem: Which benefits and/or difficulties can be obtained through the application of Scrum Framework on the evolution of the development of 'Nosso Exercício' open platform? According to the model suggested by Coughlan and Coughlan (2002), the research-action method was used to describe a dynamic conducted during this work. The Scrum implantation on 'Nosso Exercício' occurred in two stages, the first one, aimed to do a training about Scrum and the technological tools used in the development of 'Nosso Exercício'. The second one, dealt with the development of functionalities to this website. The analyses of the results showed many benefits gotten with the implantation of Scrum in this project, like focus and commitment of 'Time' during its work, the attendance to the customer's real needs (Product Owner), the framework flexibility in adjusting to environment and work conditions of each 'Time' created, the continuous learning of the software and of the process resulting of the discussions taken on planning, reviewing and retrospecto meetings. The critical work point was about the immaturity with the technologies used for the website development. The learned lessons with this research indicate that it's possible to get benefits with the implantation of Scrum Framework which overcome the difficulties, as long as the needed analysis of the environment where it was adopted be done.

Key words: Software Engineering. Agile Methodology. Scrum Framework. Software Development. Research-Action.

LISTA DE ILUSTRAÇÕES

Figura 1- Engenharia de <i>Software</i> em Camadas.....	24
Figura 2 - Modelo Cascata	25
Figura 3 - Modelo Incremental.....	26
Figura 4 - Eventos do Scrum	30
Figura 5 - Ciclo do Scrum	32
Figura 6 - Quadro <i>Kanban</i> no início de um <i>Sprint</i>	35
Figura 7 - Quadro <i>Kanban</i> com um <i>Sprint</i> em andamento.....	35
Figura 8 - Quadro <i>Kanban</i> no final de um <i>Sprint</i>	36
Figura 9 - Tela inicial do Nosso Exercício antes do desenvolvimento deste projeto.....	37
Figura 10 - Tecnologias nas quais o Drupal se apoia.....	40
Figura 11 - Ciclo da pesquisa-ação	44
Figura 12 - Quadro <i>Kanban</i> utilizado na Etapa 1 do projeto.....	56
Figura 13 - Exemplo de história de usuário utilizada no projeto	57
Figura 14 - Quadro <i>Kanban</i> Digital.....	57
Figura 15- Tela de cadastro de exercícios (História 1- Etapa 1)	60
Figura 16 - Tela de pesquisa de exercícios (História 2- Etapa 1).....	62
Figura 17 - Busca por exercícios antes da implementação da história 7.....	71
Figura 18 - Quadro no Trello indicando o início do <i>Sprint 1</i>	73
Figura 19 - Detalhes do cartão da história 7 no Trello	73
Figura 20 - Busca por exercícios após implementação da história 7	74
Figura 21 - Tela do Nosso Exercício com a opção de buscar exercícios apenas ‘Com Resolução’	76
Figura 22 - <i>Layout</i> do Nosso Exercício antes do desenvolvimento da história 7.....	94
Figura 23 - <i>Layout</i> do Nosso Exercício após o desenvolvimento da história 7	94
Figura 24 - <i>Layout</i> do Nosso Exercício antes do desenvolvimento da história 9.....	96
Figura 25 - <i>Layout</i> do Nosso Exercício depois do desenvolvimento da história 9	96
Figura 26 - <i>Layout</i> do Nosso Exercício antes do desenvolvimento da história 6.....	98
Figura 27 - <i>Layout</i> do Nosso Exercício após o do desenvolvimento da história 6	99
Figura 28 - Botão para realizar <i>download</i> de uma lista em PDF.....	101
Figura 29 - Lista de exercícios disponível para baixar ou imprimir.....	101
Figura 30 - Adicionar exercício antes do desenvolvimento da história 2	103
Figura 31 - Tela do <i>site</i> com as opções para criar uma nova lista.....	104

Figura 32 - Trecho de uma lista criada com as opções de 'Gerar Novamente' e 'Aceitar Lista'	104
Figura 33 - Exercício com as opções 'Denunciar erro' e 'Denunciar infração'	106
Figura 34 - Telas para confirmação da opção 'Denunciar erro'	106
Figura 35 - Busca pelo termo Matemática antes do desenvolvimento da história 10	108
Figura 36 - Busca pelo termo Matemática após o desenvolvimento da história 10	108
Figura 37 - Cadastro de um novo exercício com o campo 'Gabarito'	110
Figura 38 - Exercício sugerido para um novo simulado criado	111
Figura 39 - Campo 'Resposta' de uma questão do Simulado	112
Figura 40 - Resultado do simulado gerado	113
Quadro 1 - Exemplo de história de usuário	33
Quadro 2 - Modelo de história de usuário utilizado no projeto	33
Quadro 3 - Etapas do Ciclo de Condução	45
Quadro 4 - Associação do ciclo da pesquisa-ação ao ciclo de desenvolvimento dos <i>Sprints</i> no <i>Scrum</i>	46
Quadro 5 - Esquema básico da pesquisa ação aplicada neste projeto	47
Quadro 6 - Formação da primeira equipe de trabalho <i>Scrum</i> no Nosso Exercício	51
Quadro 7 - Funcionalidades existentes e desejáveis para o <i>website</i> Nosso Exercício	51
Quadro 8 - Primeira versão criada do <i>Product Backlog</i>	52
Quadro 9 - Usuários do Nosso Exercício e suas permissões antes deste projeto	52
Quadro 10 - Nível de conhecimento/experiência de cada papel nas áreas do saber necessárias ao projeto	53
Quadro 11 - Planejamento Inicial (Etapa 1)	55
Quadro 12 - Segundo Planejamento (Etapa 1)	58
Quadro 13 - História de Usuário do <i>Sprint 1</i>	59
Quadro 14 - História de Usuário do <i>Sprint 2</i>	61
Quadro 15 - História de Usuário do <i>Sprint 3</i>	63
Quadro 16 - Planejamento realizado (Etapa 1)	64
Quadro 17 - Adaptações do <i>Scrum</i> feitas na realidade do Nosso Exercício	65
Quadro 18 - Planejamento Inicial da Etapa 2 do projeto	69
Quadro 19 - <i>Product Backlog</i> utilizado na Etapa 2 do projeto	70
Quadro 20 - Descrição da história de usuário 7- <i>Sprint 1</i> (Etapa 2)	71
Quadro 21 - Critério de Aceitação da história de usuário 7	72
Quadro 22 - Teste de Aceitação da história de usuário 7	72

Quadro 23 - Descrição da história de usuário 9 - <i>Sprint 1</i> (Etapa 2).....	74
Quadro 24 - Critério de Aceitação da história de usuário 9 - <i>Sprint 1</i> (Etapa 2)	75
Quadro 25 - Teste de Aceitação da história e usuário 9- <i>Sprint 1</i> (Etapa 2)	75
Quadro 26 - Visão Geral do Ciclo de Condução da Etapa 2	78
Quadro 27 - Visão geral do <i>status</i> das histórias em cada <i>Sprint</i> da Etapa 2	80
Quadro 28 - Versão final do <i>Product Backlog</i> após o desenvolvimento do projeto	82
Quadro 29 - Comparação do uso do <i>Scrum</i> nas etapas 1 e 2 deste projeto	83
Quadro 30 - Histórias desenvolvidas em cada <i>Sprint</i> da Etapa 2.....	92
Quadro 31 - Descrição da História 7 (<i>Sprint1</i> - Etapa2)	93
Quadro 32 - Critério de Aceitação- História7 (<i>Sprint 1</i> - Etapa2).....	93
Quadro 33 - Teste de Aceitação- História7 (<i>Sprint 1</i> - Etapa2)	93
Quadro 34 - Descrição da História 9 (<i>Sprint1</i> - Etapa2)	95
Quadro 35 - Critério de Aceitação- História 9 (<i>Sprint 1</i> - Etapa2).....	95
Quadro 36 - Teste de Aceitação- História 9 (<i>Sprint 1</i> - Etapa2)	95
Quadro 37 -Descrição da História 6 (<i>Sprint2</i> - Etapa2)	97
Quadro 38 - Critério de Aceitação- História 6 (<i>Sprint 2</i> - Etapa2).....	97
Quadro 39 - Teste de Aceitação- História 6 (<i>Sprint 2</i> - Etapa2)	98
Quadro 40 - Descrição da História 1 (<i>Sprint2</i> - Etapa2)	99
Quadro 41- Critério de Aceitação- História 1 (<i>Sprint 2</i> - Etapa2).....	100
Quadro 42 - Teste de Aceitação- História 1 (<i>Sprint 2</i> - Etapa2)	100
Quadro 43 - Descrição da História 2 (<i>Sprint4</i> - Etapa2)	102
Quadro 44 - Critério de Aceitação- História 2 (<i>Sprint 4</i> - Etapa2).....	102
Quadro 45 - Teste de Aceitação- História 2 (<i>Sprint 4</i> - Etapa2)	103
Quadro 46 - Descrição da História 12 (<i>Sprint 4</i> - Etapa2)	105
Quadro 47 - Critério de Aceitação- História 12 (<i>Sprint 4</i> - Etapa2).....	105
Quadro 48 - Teste de Aceitação- História 12 (<i>Sprint 4</i> - Etapa2)	105
Quadro 49 - Descrição da história 10 (<i>Sprint 5</i> - Etapa 2)	107
Quadro 50 - Critério de Aceitação para a história de usuário 10	107
Quadro 51 - Testes de Aceitação para a história de usuário 10.....	107
Quadro 52 - Descrição da história 3 (<i>Sprint 5</i> - Etapa 2)	109
Quadro 53 - Critério de Aceitação para a história de usuário 3	109
Quadro 54 - Testes de Aceitação para a história de usuário 3- Critério 1	109
Quadro 55 - Testes de Aceitação para a história de usuário 3- Critério 2.....	111
Quadro 56 - Testes de Aceitação para a história de usuário 3- Critério 3.....	112

LISTA DE SIGLAS

BC&T	Bacharelado em Ciência e Tecnologia
CASE	<i>Computer-Aided Software Engineering</i>
CMS	<i>Content Management System</i>
CSS	<i>Cascading Style Sheet</i>
FDD	<i>Feature Driven Development</i>
IEEE	<i>Institution of Electrical and Electronic Incorporated Engineers</i>
HTML	<i>Hyper Text Markup Language</i>
PET	Programa de Educação Tutorial
PHP	<i>Hypertext Preprocessor PHP</i>
PO	<i>Product Owner</i>
PSP	<i>Personal Software Process</i>
RUP	<i>Rational Unified Process</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structure Query Language</i>
TAS	Tecnologia Ambiente e Sociedade
TSP	<i>Team Software Process</i>
UFVJM	Universidade Federal dos Vales do Jequitinhonha e Mucuri
XP	<i>Extremme Programming</i>

SUMÁRIO

1 INTRODUÇÃO	19
2 REVISÃO BIBLIOGRÁFICA	23
2.1 Engenharia de Software	23
2.1.1 Abordagem Tradicional de Desenvolvimento de Software	26
2.1.2 Abordagem Ágil de Desenvolvimento de Software.....	27
2.2 Scrum: Origem e Princípios.....	28
2.2.1 Papéis: o Time Scrum.....	29
2.2.2 Artefatos do Scrum	29
2.2.3 Eventos do Scrum	30
2.2.4 Técnicas Complementares ao Scrum.....	32
3 DESCRIÇÃO DA PLATAFORMA ABERTA NOSSO EXERCÍCIO	37
3.1 Ferramentas Computacionais utilizadas no desenvolvimento do <i>website</i> Nosso Exercício	38
4 METODOLOGIA.....	43
4.1 Classificação da pesquisa	43
4.2 Pesquisa-Ação	43
5 DESCRIÇÃO DA PESQUISA-AÇÃO	49
5.1 Fase Preliminar – Contexto geral da pesquisa.....	49
5.2 Etapa 1: Uso do <i>Scrum</i> na capacitação de alunos em desenvolvimento de <i>software</i> ..	53
5.2.1 Fase Preliminar da Etapa 1	53
5.2.2 Ciclo de Condução: Etapa 1.....	54
5.2.3 Meta-fase: Etapa 1.....	65
5.3 Etapa 2: Uso do <i>Scrum</i> no desenvolvimento do Nosso Exercício	67
5.3.1 Fase Preliminar da Etapa 2	67
5.3.2 Ciclo de Condução: Etapa 2.....	68
5.3.3 Meta-fase: Etapa 2.....	79
5.4 Comparação dos principais resultados das Etapas 1 e 2	82
6 CONSIDERAÇÕES FINAIS.....	85

REFERÊNCIAS.....	88
APÊNDICE A – CICLO DE CONDUÇÃO DA ETAPA 2: <i>SPRINT 1</i>.....	92
APÊNDICE B – CICLO DE CONDUÇÃO DA ETAPA 2: <i>SPRINT 2</i>.....	97
APÊNDICE C – CICLO DE CONDUÇÃO DA ETAPA 2: <i>SPRINT 4</i>.....	102
APÊNDICE D – CICLO DE CONDUÇÃO DA ETAPA 2: <i>SPRINT 5</i>.....	107

1 INTRODUÇÃO

O desenvolvimento de *software*¹ é uma atividade complexa e representa um desafio para a comunidade de desenvolvedores que se esforça continuamente para entregar produtos fáceis, rápidos e de qualidade, dentro do tempo e custos previamente definidos. (PRESSMAN, 2011).

A empresa americana *Standish Group*, fundada em 1985, presta consultoria para organizações que desejam melhorar os seus investimentos em projetos de *software*, usando como principal atrativo um banco de dados de pesquisas realizadas em mais de 50.000 projetos. O seu estudo mais conhecido, o relatório *Chaos Report*, divulgado a cada dois anos, desde 1994, tem evidenciado ao longo do tempo os altos índices de falhas nos projetos de desenvolvimento de *software* em todo o mundo. Este relatório expõe conclusões sobre os principais fatores de sucesso e fracasso nestes tipos de projetos, bem como as alternativas encontradas para reduzir as falhas nos mesmos. (STANDISH GROUP, 2017).

A pesquisa feita em 2014 por esta instituição em empresas de pequeno, médio e grande porte, concluiu que 17% dos projetos foram cancelados antes de serem finalizados; 55% atrasaram, estouraram o orçamento e/ou entregaram menos funcionalidades para o cliente do que o planejado; 28% foram entregues no prazo, dentro do orçamento e atendendo ao escopo completo. (STANDISH GROUP, 2014). Somando-se as duas primeiras estatísticas apresentadas, o que resulta em 72% dos projetos, observa-se um índice maior de insucesso do que sucesso, nos *softwares* desenvolvidos.

Vale ressaltar que estes números, apesar de não refletirem o cenário ideal, representam uma evolução desde os primórdios do desenvolvimento de *software*. Um dos fatores atribuídos a este progresso consiste no uso de métodos, processos e ferramentas provenientes dos princípios da Engenharia de *Software*, “uma disciplina de engenharia relacionada com todos os aspectos da produção de *software* [...]”. SOMMERVILLE (2007, p. 5).

Dois importantes abordagens comumente utilizadas em projetos de *software*, provenientes da Engenharia de *Software*, consistem no desenvolvimento tradicional e ágil. A primeira reflete um processo pesado, que se caracteriza pelo foco em planos detalhados definidos no princípio do projeto, documentação extensa e uso de processos cada vez mais complexos. (SABBAGH, 2016). A segunda, considerada leve, prevê uma entrega mais rápida

¹ [...] qualquer tipo de programa, aplicativo, utilitário, arquivo, sistema operacional e jogos. (FIALHO JR., 2002, p. 173).

do *software* para o cliente, envolve pequenas equipes auto-organizadas que utilizam métodos informais e focam na entrega operacional de cada parte do *software* a ser desenvolvido. (PRESSMAN, 2006).

Não existe uma única abordagem que atenda às necessidades de todos os tipos de sistemas e sim, aquela que se adequa melhor a uma ou outra realidade de projeto de *software*. Sommerville (2007, p. 262) explica que os modelos ágeis “foram criados principalmente para apoiar o desenvolvimento de aplicações de negócios nas quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento”, como por exemplo, as aplicações *web*². Diferentemente dos *softwares* de aplicações convencionais (não *web*) que tendem a evoluir através de uma série de versões planejadas e cronologicamente espaçadas, as aplicações *web* evoluem continuamente. (PRESSMAN, 2006).

Métodos baseados em desenvolvimento ágil surgiram como uma opção ao tradicional e, devido a sua flexibilidade e simplicidade, têm sido amplamente utilizados na criação de aplicações *web*. Neste contexto destaca-se o *Scrum*, um método ágil empírico, que, conforme define seus criadores, *Ken Schwaber* e *Jeff Sutherland*, consiste num “*Framework*”³ dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.” (SCHWABER; SUTHERLAND, 2016, p. 3).

Em sua 11ª edição, o relatório anual Estado do Ágil, elaborado pela *Version One*⁴, maior pesquisa sobre cultura ágil do mundo realizada em 2016, divulgou o *Scrum* como o método ágil mais utilizado conforme relato de 58% dos seus entrevistados. (VERSIONONE, 2016). De fato, a abordagem enxuta para criação de produtos proposta pelo *Scrum* tem atraído o interesse de diversas organizações e equipes de trabalho de desenvolvimento de *software*.

Com base no exposto acima, este estudo descreve o processo de implantação e análise do *Framework Scrum* no desenvolvimento do *website* Nosso Exercício. Esta aplicação *web* consiste em um dos projetos do Programa de Educação Tutorial, PET-UFVJM/Campus do Mucuri situada na cidade de Teófilo Otoni-MG e tem como finalidade o compartilhamento aberto de exercícios didáticos de diversas áreas do conhecimento.

² Termo empregado para identificar a internet. (FIALHO JR, 2002. p.189).

³ [...] estrutura básica que pretende servir de suporte e guia para a construção [...] de algo com uso prático. (SABBAGH, 2016, p. 33).

⁴ Empresa americana pioneira no uso e comercialização de ferramentas de gerenciamento ágil. (VERSIONONE, 2017).

Assim, o objetivo geral desta pesquisa consiste em identificar quais benefícios e/ou dificuldades podem ser obtidas através da aplicação do *Framework Scrum* na evolução do desenvolvimento da plataforma aberta Nosso Exercício.

Antes desta análise, o projeto da plataforma aberta Nosso Exercício encontrava-se em sua primeira versão, com algumas funcionalidades já em operação e nenhum princípio básico da Engenharia de *Software* havia sido empregado formalmente no seu desenvolvimento. Isto quer dizer que as atividades de criação da mesma focaram apenas na programação do *software*, desconsiderando qualquer tipo de controle, planejamento e/ou possível evolução da aplicação. Seguindo a dinâmica evolutiva das aplicações *web*, novas funcionalidades foram demandadas para o Nosso Exercício por solicitação de seus idealizadores, sendo um deles, professor membro do Programa do Mestrado Profissional em Tecnologia Ambiente e Sociedade (TAS) da UFVJM/Campus Mucuri.

Após o contato inicial com a realidade do *software* a ser desenvolvido foi possível identificar que, apesar da necessidade de evolução, não existiam para este projeto metas claras nem um plano de trabalho a ser seguido. Os requisitos⁵ não estavam formalizados e a equipe disponível não estava madura nas tecnologias utilizadas.

Levando-se em consideração o desafio enfrentado na geração de *software* de qualidade já apresentado e o limite de tempo disponível para a realização desta pesquisa, foi proposto, além das atividades de desenvolvimento do *software web* em questão, o uso de um processo da Engenharia de *Software* com intuito de se obter maior controle e qualidade do produto final a ser desenvolvido. O método ágil *Scrum* foi o escolhido para gerenciar as atividades realizadas durante a execução do projeto.

As estratégias adotadas para que o objetivo geral fosse alcançado se configuraram nos seguintes objetivos específicos: realizar treinamento para a equipe de desenvolvimento, tanto das tecnologias adotadas quanto do *framework* proposto; levantar e registrar novos requisitos para a plataforma; planejar as etapas de desenvolvimento; desenvolver e testar as funcionalidades levantadas; analisar os resultados com intuito de aprimorar o processo adotado.

Seguindo o modelo sugerido por Coughlan e Coughlan (2002), o método de pesquisa-ação foi utilizado para descrever a dinâmica conduzida durante este trabalho. A implantação do *Scrum* no Nosso Exercício ocorreu em duas etapas, a primeira, objetivou realizar uma capacitação sobre o *Scrum* e as ferramentas tecnológicas utilizadas no

⁵São descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. [...] refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema[...]. (SOMMERVILLE, 2007, p.79).

desenvolvimento do Nosso Exercício. Já a segunda, tratou do desenvolvimento das funcionalidades para este *website*.

2 REVISÃO BIBLIOGRÁFICA

2.1 Engenharia de *Software*

O termo ‘Crise do *Software*’ despontou em meados de 1968 chamando a atenção para a forma imatura (artesanal e informal) como o *software* vinha sendo produzido até então. Esta época foi caracterizada por avanços tecnológicos do *hardware*, aumentando exponencialmente a sua capacidade de processamento e armazenamento, demandando, conseqüentemente, soluções de *software* mais complexas capazes de explorar o poder computacional existente. (SOMMERVILLE, 2007).

Novas técnicas de planejamento e projeto para o *software* foram necessárias impulsionando o surgimento da Engenharia de *Software* como uma proposta de solução para estes problemas, estabelecendo métodos sistemáticos, disciplinados e quantificáveis para o trabalho de criação de *software* conforme definido pelo *Institution of Electrical and Electronic Incorporated Engineers (IEEE)* em 1993. (PRESSMAN, 2006).

Um conceito bem detalhado de Engenharia de *Software* é apresentado por Brod, como sendo uma:

Área da computação voltada à especificação, ao desenvolvimento e à manutenção de sistemas de *software*, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando à organização, produtividade e qualidade. Atualmente, essas tecnologias e práticas englobam linguagens de programação, banco de dados, ferramentas, plataformas, bibliotecas, padrões, processos e a questão da qualidade de *software*. Os fundamentos científicos para a engenharia de *software* envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de *software*, avaliando e garantindo suas qualidades. Além disso, a engenharia de *software* deve oferecer mecanismos para se planejar e gerenciar processo de desenvolvimento de um sistema computacional. (BROD, 2015, p.193).

De forma mais objetiva, Sommerville (2007, p. 5) define Engenharia de *Software* como “uma disciplina de engenharia relacionada com todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que este entrar em operação”. Através destes princípios, os engenheiros de *software* são apoiados por ferramentas, métodos e processos fundamentais às suas atividades, buscando criar soluções que atendam às necessidades dos seus clientes.

A Figura 1 apresenta a Engenharia de *Software* como uma tecnologia em camadas. Os processos oferecem controle gerencial aos projetos de *software* para que aconteçam dentro de prazos e custos pré-estabelecidos. Os métodos incluem tarefas que devem ser realizadas durante as etapas de desenvolvimento do *software*. As ferramentas fornecem apoio

automatizado ou semiautomatizado para os processos e métodos. A base em que todos se apoiam é o foco na qualidade. (PRESMAN, 2006).

Figura 1 - Engenharia de *Software* em Camadas



Fonte: PRESSMAN, 2006.

As tarefas de desenvolvimento estabelecidas pelos métodos são distribuídas em algumas etapas. Teles as relaciona da seguinte maneira:

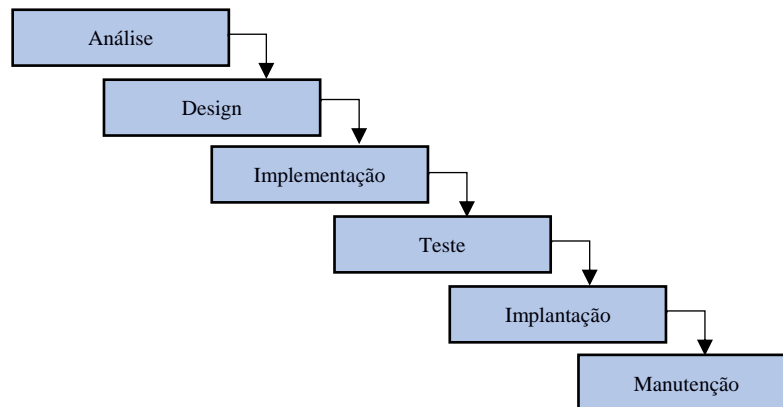
1. Análise - a equipe faz o levantamento dos requisitos e busca compreendê-los detalhadamente.
2. *Design* - com base na análise a equipe projeta a arquitetura do sistema.
3. Implementação - a equipe se baseia na arquitetura e na análise para implementar as diversas partes do *software*.
4. Teste - para verificar se o sistema atende às necessidades especificadas pelo usuário, a equipe testa o *software* e faz as correções necessárias.
5. Implantação - o sistema é colocado em produção e os usuários finais passam a utilizá-lo.
6. Manutenção - até o fim da sua vida, o *software* poderá sofrer alterações por diversas razões, tais como correções e inclusão de novas funcionalidades. (TELES, 2006, p. 21).

As ferramentas automatizam muitas das tarefas necessárias no desenvolvimento de *software*. Quando podem ser utilizadas de forma integrada de modo que a informação criada por uma pode ser usada por outra, são chamadas de ferramentas *Computer-Aided Software Engineering* _ Engenharia de *Software* Apoiada por Computador (CASE). (PRESSMAN, 2006). Este tipo de ferramenta induz “um alto grau de rigidez e padronização em todo o processo de desenvolvimento de sistemas”. (STAIR, 2002, p. 384).

Um processo abrange um conjunto de atividades que levam à produção de um produto de *software*. (SOMMERVILLE, 2007). O uso de um processo inicia pela escolha de um modelo de ciclo de vida. De acordo com a NBR ISO/IEC 12207 (1998), o ciclo de vida é a “estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de *software*, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso”. Aqui serão abordados dois modelos de ciclo de vida considerados essenciais para o entendimento da proposta deste trabalho: Cascata e Incremental.

O modelo Cascata (Figura 2), mais antigo e ainda utilizado, estabelece uma sequência linear para o desenvolvimento de *software* de forma que todo o trabalho seja realizado em apenas um ciclo. De acordo com este modelo uma fase só começa quando a anterior termina.

Figura 2 - Modelo Cascata



Fonte: TELES, 2006.

Pressman (2006) relata alguns problemas com este modelo:

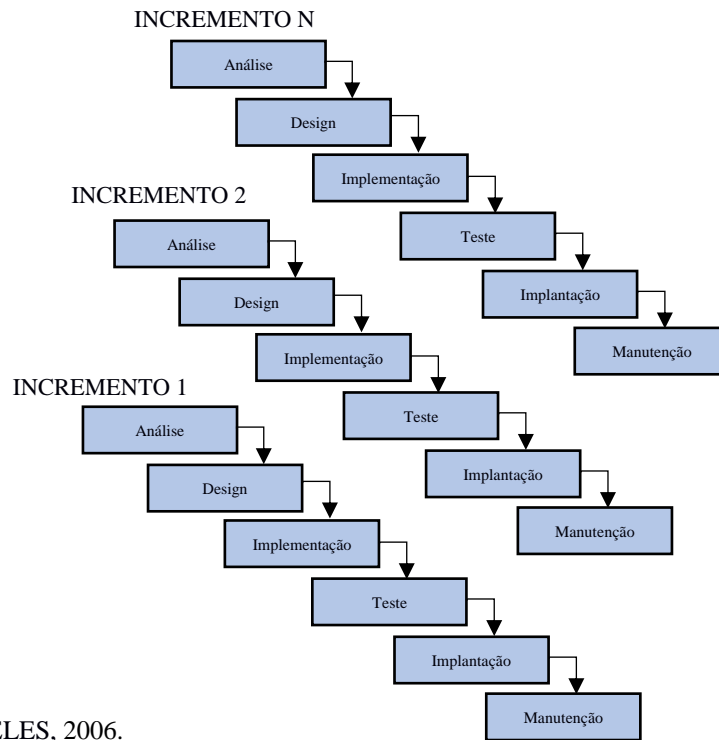
1. O fluxo sequencial estabelecido não condiz com a realidade da maior parte dos projetos de *software*.
2. Como o ciclo é único, cada etapa deve ser cuidadosamente planejada, pois, mudanças só poderão ocorrer num próximo ciclo. Na prática, há uma impossibilidade do cliente prever antecipadamente todos os requisitos para o *software*, como o modelo exige.
3. O tempo que o cliente precisa esperar para receber a versão executável, que se dá, apenas, após a última etapa do projeto.

O modelo Incremental (Figura 3) contempla o desenvolvimento de incrementos do *software* através da repetição de vários ciclos do modelo cascata. Um incremento é uma parte do *software* passível de ser entregue para o cliente. Através deste modelo o *software* vai sendo desenvolvido aos poucos, isto é, o primeiro incremento corresponde a uma versão com poucas funcionalidades, porém críticas para o cliente, enquanto que a última contém todas as funcionalidades do sistema. (PRESSMAN, 2006).

De acordo com Sommerville (2007), o modelo incremental tem uma série de vantagens em relação ao cascata, como o fato do cliente poder receber uma versão do *software* mais cedo, se beneficiando dele. Além disso, as falhas do projeto diminuem uma vez que a cada

incremento entregue, o cliente tem a oportunidade de dar o seu *feedback* e a equipe de desenvolvimento agir sobre elas.

Figura 3 - Modelo Incremental



Fonte: TELES, 2006.

Teles (2006) usa o termo ‘desenvolvimento tradicional’ para projetos que utilizam como base o modelo cascata e ‘desenvolvimento ágil’, para aqueles que se baseiam no modelo incremental para o desenvolvimento de *software*. O autor menciona, porém, exemplos que podem fugir a esta regra, como o caso do *Rational Unified Process* (RUP) que, apesar de propor a construção do *software* de forma incremental, é considerado um método tradicional, devido a linearidade com que propõe esta atividade.

2.1.1 Abordagem Tradicional de Desenvolvimento de Software

Duas importantes abordagens propostas pela Engenharia de *Software* para o desenvolvimento de sistemas computacionais são o ‘processo tradicional’ (ou modelo prescritivo) e o ‘processo ágil’, descrito na próxima seção. Em relação ao primeiro, Somerville relata que:

Nas décadas de 1980 e início de 1990, havia uma visão geral que a melhor maneira de obter o *software* era por meio de um cuidadoso planejamento de projeto, garantia de qualidade formalizada, uso de métodos de análise e projeto apoiados por

ferramentas CASE e controlados por um rigoroso processo de desenvolvimento de *software*. Essa visão vem essencialmente, da comunidade de engenharia de *software* preocupada com sistemas de *software* grandes e de longa vida, geralmente constituídos de um grande número e programas individuais. (SOMERVILLE, 2007, p. 262).

Processos tradicionais são considerados pesados por serem rígidos e burocráticos e utilizarem, na maioria das vezes, o modelo de ciclo de vida cascata. São dotados de documentação detalhada e métodos que buscam prever, senão todos, a maior parte dos requisitos necessários para a criação de um produto de *software* com qualidade. Este esforço existe para que mudanças durante o projeto sejam evitadas, pois implicam em um aumento considerável no custo e no tempo de entrega do mesmo.

Pressman esclarece:

Quando esses modelos prescritivos de processo são aplicados, o objetivo é melhorar a qualidade do sistema para tornar os projetos mais gerenciáveis, as datas de entrega e os custos mais previsíveis e para guiar equipes de engenheiros de *software* à medida que eles realizam o trabalho necessário para construir um sistema. Infelizmente, tem havido épocas em que esses objetivos não têm sido alcançados. Se os modelos prescritivos forem aplicados dogmaticamente e sem adaptação, eles podem aumentar o nível de burocracia para desenvolvedores e clientes. (PRESSMAN, 2006, p. 21).

Exemplos destes modelos são o *Rational Unified Process* (RUP), *Personal Software Process* (PSP) e *Team Software Process* (TSP). (PAULA FILHO, 2003).

2.1.2 Abordagem Ágil de Desenvolvimento de Software

A insatisfação com os métodos tradicionais deu origem aos processos ágeis ou leves, que primam pela simplicidade. Tais processos utilizam o modelo incremental, são mais flexíveis e visam a construção rápida de um produto de *software* que gere valor para o cliente. Sommerville (2007, p. 262) explica que os modelos ágeis “foram criados principalmente para apoiar o desenvolvimento de aplicações de negócios nas quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento”. São baseados em princípios, práticas e valores provenientes do Manifesto Ágil, evento que em fevereiro de 2001, nos Estados Unidos, reuniu diversos profissionais de desenvolvimento de *software* da época. As declarações deste manifesto foram:

- Indivíduos e interação entre eles mais que processos e ferramentas.
- *Software* em funcionamento mais que documentação abrangente.
- Colaboração com o cliente mais que negociação de contratos.
- Responder a mudanças mais que seguir um plano. (BECK et al., 2001).

Isso não quer dizer que os métodos ágeis dispensam o uso de processos e ferramentas ou rejeitam documentação, planejamento e contratos previstos nos modelos

tradicionais. Apenas valorizam mais as pessoas e a maneira como elas se comunicam, a rápida entrega do *software* funcionando, a participação efetiva do cliente como membro da equipe e um planejamento flexível. Exemplos destes modelos são, *Extreme Programming (XP)*, *Feature Driven Development (FDD)*, *Crystal* e *Scrum*. (SOMMERVILLE, 2007). Dentre os métodos ágeis exemplificados, o *Scrum* foi o escolhido para o desenvolvimento deste trabalho.

2.2 *Scrum*: Origem e Princípios

Apesar de comumente ser utilizado em projetos de *software*, o termo *Scrum* não surgiu nessa área do conhecimento. Foi citado inicialmente no artigo “*The new new product development game*” (O novo jogo para o desenvolvimento de novos produtos), de *Hiroataka Takeuchi* e *Ikujiro Nonaka* em 1986. No artigo, os autores analisavam as equipes que atuavam nas empresas mais produtivas e inovadoras da época: *Honda*, *Xerox*, *Hewlett-Packard*, entre outras. Estas equipes apresentavam algumas características em comum: tinham autonomia, eram multifuncionais, os líderes eram participativos e se comprometiam a retirar os obstáculos encontrados durante a realização do trabalho. Os autores japoneses compararam o comportamento destas equipes com uma jogada de *Rugby* em que o Time se reúne, formando uma unidade, visando atingir o mesmo objetivo: avançar com a bola no campo. Posteriormente, *Ken Schwaber* e *Jeff Sutherland* em 1993, associaram o termo ao desenvolvimento de *software* sendo reconhecidos como criadores do método. (SUTHERLAND, 2014).

Sabbagh apresenta uma definição resumida do *Scrum*:

Scrum é um *Framework* Ágil, simples e leve, utilizado para o desenvolvimento de produtos complexos imersos em ambientes complexos. *Scrum* é embasado no empirismo e usa uma abordagem iterativa e incremental para entregar valor com frequência, assim, reduzindo os riscos do projeto. (SABBAGH, 2016, p. 19).

Scrum é considerado um *framework* e não uma metodologia. Um *framework* contém apenas a estrutura básica que servirá de suporte para a criação de um produto. O *Scrum* não entra em detalhes sobre como o Time deve desenvolver um determinado produto ou qual técnica utilizar para compreender as necessidades de um cliente, nem mesmo como lidar com estes. Há uma compreensão de que cada projeto é único e que as práticas necessárias para o sucesso de cada um são muito específicas à sua realidade. (SABBAGH, 2016).

O Guia do *Scrum*, material produzido e mantido pelos criadores do *Scrum*, apresenta os seus três pilares: transparência, inspeção e adaptação. A transparência se baseia na visibilidade, ou seja, todos os envolvidos no projeto devem ter uma visão comum e clara do processo. As inspeções são realizadas para avaliar se o que está sendo feito pela equipe está

direcionado às metas. Já as adaptações são essenciais para que os ajustes necessários sejam feitos. (SCHWABER; SUTHERLAND, 2016).

2.2.1 Papéis: o Time Scrum

Conforme descreve Brod (2015), apenas três papéis são assumidos em um Time Scrum:

- **Product Owner**: o dono do projeto, o cliente, aquele que está pagando por ele, ou, dentro de uma empresa ou instituição, é o responsável por sua entrega.

- **Scrum Master**: é um membro da equipe que garante o bom andamento do projeto, assegurando que seus ritos sejam cumpridos, seus artefatos sejam usados de maneira correta e qualquer obstáculo ao trabalho da equipe seja removido.

- **Time de Desenvolvimento (Equipe)**: é quem realiza as ações de construção do projeto. Não há hierarquia entre os membros da equipe, e ela se auto gerencia na distribuição de tarefas e no agendamento da codificação das histórias (funcionalidades) do *software*.

Em um Time Scrum as pessoas que desempenham estes papéis devem ser comprometidas, pois, são igualmente responsáveis pelos resultados do trabalho, para isso trabalham de forma colaborativa para alcançarem seus resultados. (SABBAGH, 2016).

2.2.2 Artefatos do Scrum

Quatro são os artefatos do Scrum definidos no Guia do Scrum (SCHWABER; SUTHERLAND, 2016):

- **Product Backlog ou Backlog do Produto**: é uma lista ordenada de tudo que deve ser necessário no produto. O *Product Owner* é responsável pelo *Backlog* do Produto, incluindo seu conteúdo, disponibilidade e ordenação. Esta lista nunca está completa, devendo acompanhar a dinâmica de evolução do produto e do ambiente no qual ele será utilizado. O *Backlog* do Produto lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões. Os itens do *Backlog* do Produto possuem os atributos de descrição, ordem, estimativa e valor.

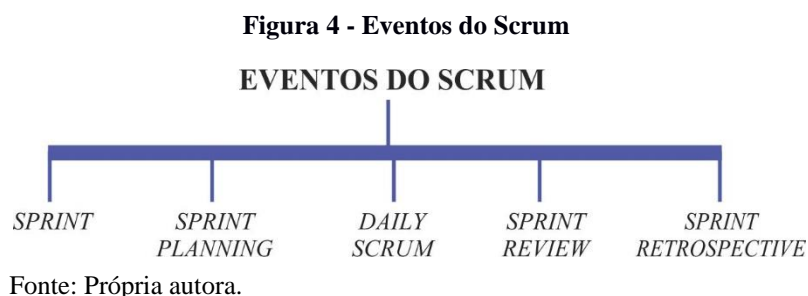
- **Sprint Backlog ou Backlog do Sprint**: é um conjunto de itens do *Backlog* do Produto selecionados para serem trabalhados dentro de um *Sprint*, juntamente com o plano para entregar o incremento do produto e atingir o objetivo do *Sprint*.

- **Definição de ‘Pronto’:** indica para o Time quando um item do *Backlog* do Produto ou um incremento está ‘Pronto’. Em um Time *Scrum* todos devem ter o mesmo entendimento de quando o trabalho está completo, ou seja, o significado de ‘Pronto’. “A definição de ‘Pronto’ varia de time para time, de projeto para projeto”. (SABBAGH, 2016, p. 155).

- **Incremento do Produto:** o incremento é a soma de todos os itens do *Backlog* do Produto já completados durante os *Sprints*. Ao final de cada *Sprint* um novo incremento deve estar ‘Pronto’, o que significa que deve estar na condição utilizável pelo *Product Owner*.

2.2.3 Eventos do Scrum

Eventos no *Scrum* correspondem ao seu próprio ciclo de desenvolvimento. Projetos *Scrum* seguem um ciclo de desenvolvimento incremental, que, por sua vez, é iterativo. Por iterativo entende-se que estes ciclos se repetem sucessivamente, com todas as atividades necessárias para se gerar uma parte do produto estável e funcionando. (SABBAGH, 2016, p.41). A Figura 4 esquematiza os eventos do *Scrum*.



Os ciclos de desenvolvimento no *Scrum* são divididos em tempos menores denominados *Sprints*. “O coração do *Scrum* é o *Sprint*, um *time-box* de um mês ou menos, durante o qual um ‘Pronto’, versão incremental potencialmente utilizável do produto, é criado”. (SCHWABER; SUTHERLAND, 2016, p.8). No decorrer de um projeto com *Scrum* os *Sprints* vão sendo executados, um após o outro, sem intervalos, tornando-se cada vez mais completo, até ser finalizado. (SABBAGH, 2016).

Os *Sprints* são compostos por uma *Sprint Planning* do *Sprint* (*Sprint Planning*), reuniões diárias (*Daily Scrum*), o trabalho de desenvolvimento, uma revisão do *Sprint* (*Sprint Review*) e a retrospectiva do *Sprint* (*Sprint Retrospective*). (SCHWABER; SUTHERLAND, 2016).

É na *Sprint Planning* do *Sprint* que a equipe planeja o ciclo de desenvolvimento do *Sprint* atual. Esta reunião acontece no primeiro dia do *Sprint*. Nela devem estar presentes o *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*. Neste momento a meta do *Sprint* é definida, *Product Owner* e o Time negociam quais itens da lista serão desenvolvidos durante o *Sprint* e um plano de trabalho é realizado. O conjunto de itens do *Sprint* e o seu planejamento é denominado *Sprint Backlog*. (SABBAGH, 2016).

O *Daily Scrum* é uma reunião curta (sugere-se um tempo máximo de 15 minutos), realizada diariamente pelo Time de Desenvolvimento com intuito de responder a três perguntas:

- O que eu fiz desde a última reunião?
- O que pretendo fazer até a próxima reunião?
- Que impedimentos estão em meu caminho?

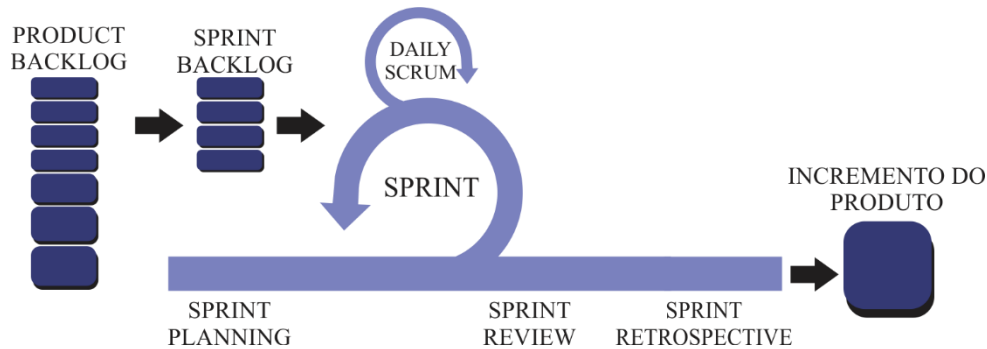
As respostas destas perguntas promovem a comunicação do Time, dão visibilidade ao trabalho, auxiliam na remoção de obstáculos e no cumprimento da meta do *Sprint*. O *Product Owner* não participa desta reunião, enquanto o *Scrum Master* pode participar como um facilitador, ajudando a equipe a manter o foco e não ultrapassar o tempo previsto. (SABBAGH, 2016).

No encerramento do *Sprint* ocorrem duas reuniões, a *Sprint Review* (Reunião de Revisão) e a *Retrospective Review* (Reunião de Retrospectiva). A *Sprint Review* objetiva obter o *feedback* do cliente sobre o incremento do produto desenvolvido no *Sprint*. Além do *Product Owner*, outros representantes do cliente podem participar deste momento, bem como o Time de Desenvolvimento e o *Scrum Master*. Através desta reunião é possível diminuir os riscos do projeto, pois o Time, através do *feedback* do cliente poderá fazer os ajustes necessários, corrigindo os erros, atendendo melhor às necessidades do cliente. (SABBAGH, 2016).

Por fim, o último evento do *Scrum* é denominado *Retrospective Sprint*. Esta reunião acontece após a *Sprint Review* e deve contar com a participação do Time de Desenvolvimento, *Product Owner* e *Scrum Master*. Neste momento os integrantes têm a oportunidade de aprenderem com seus acertos e erros durante o último *Sprint*. Tudo é inspecionado, os processos de trabalho utilizados, o comportamento das pessoas, os relacionamentos, as ferramentas, o ambiente. Assim, busca-se manter o que deu certo e melhorar o que não deu para os próximos *Sprints*. A partir daí um novo ciclo se inicia, considerando novos itens do *Product Backlog* ou ainda contemplando as mudanças solicitadas pelo *Product Owner* no *Sprint* anterior. O ciclo permanece até que o *software* seja totalmente concluído. (SABBAGH, 2016).

A Figura 5 apresenta o ciclo completo do *Scrum*.

Figura 5 - Ciclo do Scrum



Fonte: SABBAGH, 2016, p. 42.

2.2.4 Técnicas Complementares ao Scrum

Nesta seção serão abordadas algumas técnicas complementares ao *Scrum*, oriundas de outros métodos ágeis, utilizadas no desenvolvimento deste trabalho, a saber: *User Story* ou história de usuário, Critérios e Teste de Aceitação e quadro *Kanban*.

Uma história de usuário (*User Story*) representa uma funcionalidade do sistema, ou seja, contém a descrição de uma necessidade do usuário do produto sob o seu ponto de vista. Originam-se do *Extremme Programming* (XP), processo de desenvolvimento de *software* voltado para projetos cujos requisitos são vagos e estão em constante mudança. É comumente indicado para equipes pequenas de até 12 desenvolvedores, usa a orientação a objetos e um modelo de ciclo de vida incremental. (TELES, 2006). Ainda segundo este autor, XP:

[...] busca assegurar que o cliente receba o máximo de valor de cada dia de trabalho da equipe de desenvolvimento. Ele é organizado em torno de um conjunto de valores e práticas que atuam de forma harmônica e coesa para assegurar que o cliente sempre receba um alto retorno do investimento em *software*. (TELES, 2006, p. 21).

No *Scrum*, o uso das histórias de usuário é opcional. Cada item do *Product Backlog* que será desenvolvido pode ser representado no formato de uma história de usuário. Recomenda-se que sejam escritas pelo próprio cliente (*Product Owner*) e refinadas para o trabalho em conjunto com o Time de Desenvolvimento. Não podem ser consideradas como suficientes para o desenvolvimento, representam apenas o começo. Devem ser seguidas por uma série de conversas entre o *Product Owner* e os membros do Time. (SABBAGH, 2016).

Uma história de usuário deve ser simples o suficiente de forma que possa ser facilmente explicada, porém completa e independente. Deve ser passível de ser desenvolvida

dentro de um *Sprint*, facilmente testável e pode ser dividida em uma série de tarefas menores. (BROD, 2015).

Um modelo proposto por Cohn (2004) estabelece o seguinte formato para a escrita de uma história de usuário:

- **Como um...** (QUEM_ define o usuário).
- **Eu quero...** (O QUÊ_ define a necessidade do usuário)
- **Para...** (POR QUÊ_ define o benefício que o usuário terá ao receber a funcionalidade desenvolvida).

O Quadro 1 mostra um exemplo de história de usuário.

Quadro 1 - Exemplo de história de usuário

	<i>Como um Comprador de livros, quero buscar um livro pelo título para escolher comprá-lo.</i>

Fonte: SABBAGH, 2016.

Teles (2006, p.70) afirma que “não devem existir regras de formato” para a escrita de uma história de usuário. O formato utilizado neste trabalho é exemplificado no Quadro 2.

Quadro 2 - Modelo de história de usuário utilizado no projeto

ID da História:	10	Prioridade:	Alta	Sprint:	5
Nome da História:	Mostrar quantidade de itens resultantes da busca.				
Descrição:	Como usuário anônimo ou registrado, ao realizar uma pesquisa, quero que o sistema retorne a quantidade de itens resultantes da busca.				
Pontos estimados:		Pontos consumidos:			

Fonte: Própria autora.

O modelo ilustrado identifica os seguintes campos:

- ID da História: estabelece um identificador para a história conforme está definido no *Product Backlog* do projeto;
- Prioridade: classifica a importância da história na visão do *Product Owner* em alta, média ou baixa;

- *Sprint*: identifica em qual *Sprint* aquela história será trabalhada;
- Descrição: Descreve a história seguindo o modelo por Cohn (2004).
- Pontos⁶ estimados: registra uma estimativa feita em pontos, o planejamento feito pela equipe para o desenvolvimento da história;
- Pontos consumidos: registra na realidade quantos pontos foram utilizados para o desenvolvimento da história.

Sabbagh (2016) explica que as histórias de usuário não devem ser consideradas suficientes para o desenvolvimento de um trabalho, elas são na verdade, um convite ao diálogo. Uma série de conversas entre o *Product Owner* e o Time deverão acontecer nas reuniões de cada *Sprint*, à fim de se obter os detalhes de negócio necessários para o desenvolvimento das funcionalidades. O autor sugere, ainda que, anexadas às histórias, detalhes referentes ao negócio podem ser satisfatoriamente documentados através de Critérios e Testes de Aceitação.

Um Critério de Aceitação permite à equipe identificar quando uma história está implementada por completo e com sucesso. Estes critérios devem ser estabelecidos pelo cliente que passa a assumir uma responsabilidade sobre aquela história (GOMES, 2013). Já os Testes de Aceitação simulam a interação do usuário com o sistema. Através de um passo-a-passo o usuário executa ações e compara os resultados com aqueles esperados para cada funcionalidade. (TELES, 2006).

Já o *Kanban*, surgiu dos sistemas de cartões usados nas indústrias de produção japonesa, que tinham como finalidade o gerenciamento do fluxo de trabalho através da organização do desenvolvimento. Não é um processo e nem descreve papéis e fases para serem seguidas, mas apresenta uma abordagem para mudança gerencial do projeto. (MARIOTTI, 2012). Na prática consiste num quadro de tarefas, geralmente dividido em três colunas: a fazer; em execução; feito. Em cada coluna cartões são afixados e vão avançando no quadro à medida que o trabalho da equipe progride. (BROD, 2015).

Cada cartão no quadro *Kanban* contém uma história de usuário (ou tarefa), sua descrição e a identificação do membro do Time de Desenvolvimento que está atuando sobre aquele cartão. Quando as atividades envolvidas com o cartão na coluna 'em execução' são finalizadas, o mesmo é movido para a coluna seguinte, liberando espaço para entrada de um novo cartão. (MARIOTTI, 2012). "O *Kanban* atua fornecendo visibilidade nos processos,

⁶ Unidade de medida usada para estimar o esforço necessário para o desenvolvimento de uma história por um desenvolvedor. O significado de um ponto pode variar de projeto para projeto, podendo, por exemplo, ser representado em horas, dias, dentre outros. (TELES, 2006).

deixando explícitos os problemas e prendendo o foco da equipe em qualidade”. (MARIOTTI, 2012, p.7).

Kanban pode ser associado ao *Scrum* para dar maior visibilidade às atividades desempenhadas dentro de um *Sprint*. As Figuras 6, 7 e 8, respectivamente, demonstram uma situação hipotética de uso do *Kanban* no início, meio e fim de um *Sprint*, onde A, B e C referem-se aos cartões que serão trabalhados. No início de um *Sprint* os cartões de histórias A, B e C foram organizados no *Kanban* na coluna ‘A fazer’, pois, ainda não foram contemplados no desenvolvimento por nenhum membro do Time (Figura 6).

Figura 6 -Quadro Kanban no início de um Sprint

PROJETO X		
A fazer	Em execução	Feito
A		
B		
C		

Fonte: Própria autora.

A Figura 7 demonstra um momento do *Sprint* em que o Time está trabalhando com os cartões de história B e C, porém, ainda não iniciaram o cartão A.

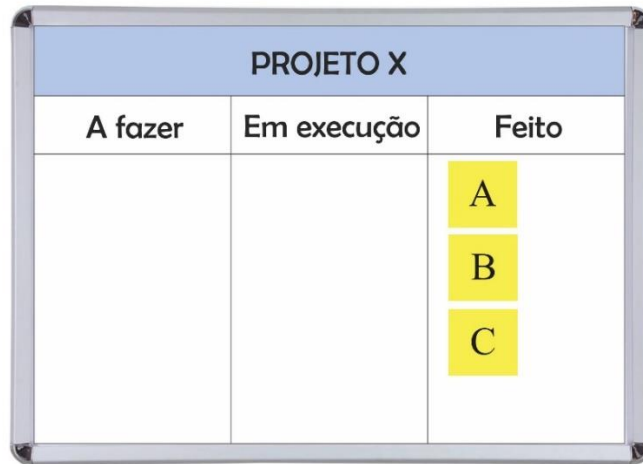
Figura 7 - Quadro Kanban com um Sprint em andamento

PROJETO X		
A fazer	Em execução	Feito
A	B	
	C	

Fonte: própria autora.

Após todos os cartões serem contemplados pelo Time de Desenvolvimento, o quadro *Kanban* assume o aspecto semelhante ao da Figura 8, indicando que todos os cartões foram trabalhados e finalizados.

Figura 8 - Quadro *Kanban* no final de um *Sprint*



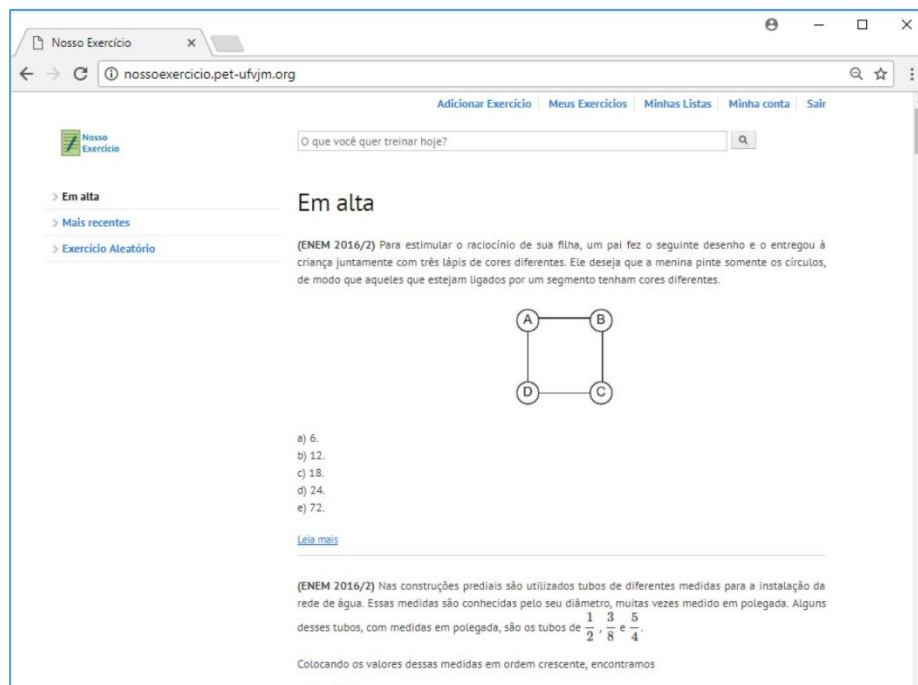
PROJETO X		
A fazer	Em execução	Feito
		A
		B
		C

Fonte: própria autora.

3 DESCRIÇÃO DA PLATAFORMA ABERTA NOSSO EXERCÍCIO

A plataforma aberta Nosso Exercício consiste num projeto do Programa de Educação Tutorial – (PET), da UFVJM - Campus do Mucuri, Teófilo Otoni- MG e tem como objetivo o compartilhamento de exercícios didáticos de diversas áreas do conhecimento e níveis de ensino, usando para isso uma tecnologia de código aberto. Atualmente existem mais de 2.370 exercícios cadastrados na plataforma. A Figura 9 mostra a página inicial do *website* Nosso Exercício antes do desenvolvimento deste trabalho.

Figura 9 - Tela inicial do Nosso Exercício antes do desenvolvimento deste projeto



Fonte: Própria autora.

Utilizando essa plataforma alunos e professores podem encontrar exercícios didáticos, com ou sem resolução, dos mais variados temas e assuntos. A ideia é que o conteúdo facilmente disponível, auxilie os alunos em seu aprendizado, ao treinar os seus estudos. Outra possibilidade é a elaboração e compartilhamento de listas de exercícios por parte dos professores e/ou alunos. A plataforma possibilita ainda que cada pessoa inscrita insira exercícios, com a resolução opcional, utilizando um *design* prático e rápido. O *site* também conta com uma barra de busca, onde, por palavras-chave, é possível localizar exercícios específicos.

Um dos propósitos do Projeto Nosso Exercício é que o uso da plataforma encoraje os estudantes a terem responsabilidade pelo seu próprio aprendizado, monitorando o seu progresso. Assim, busca incentivar também a criação de comunidades de estudantes para permitir aos alunos formarem grupos de estudos, clubes ou times de colaboração, encorajando o crescimento dos laços de amizade e parceria.

O acesso à plataforma é feito através da internet, de modo que esta atividade possa ser feita não apenas dentro da universidade, mas também em qualquer lugar que ofereça este tipo de serviço, possibilitando a interação de professor com o aluno em localidades diferentes. Como os exercícios são de acesso gratuito e via internet, tanto discente, docente ou qualquer pessoa que demonstrar interesse pode desfrutar dos exercícios. Assim a plataforma é uma disseminadora do conhecimento, alcançando e prestando serviço a toda comunidade.

O *website* Nosso Exercício democratiza o acesso à educação sem nenhum fim lucrativo, está sustentado nos pilares da comunidade DRUPAL⁷, sendo inteiramente desenvolvido com *softwares* livres⁸ e carrega essa ideologia nos seus princípios. Com a atual situação de desigualdade social do Brasil, o advento da internet e o seu fácil acesso, nota-se o quanto projetos como esses podem contribuir para a diminuição dessas disparidades, já que a Educação é a melhor maneira de ascensão social.

3.1 Ferramentas Computacionais utilizadas no desenvolvimento do *website* Nosso Exercício

O desenvolvimento de um *software* requer, além de atividades e recursos de gerenciamento, o uso de ferramentas computacionais necessárias para a transformação das solicitações do cliente em funcionalidades operacionais. Nesta seção descreve-se os principais recursos computacionais adotados no desenvolvimento do Nosso Exercício, tanto na primeira versão do *website* quanto nesta proposta de evolução.

- Linguagem de programação PHP (versão 5)

Uma linguagem de programação consiste num conjunto de símbolos e regras usados por um profissional de Sistemas de Informação para escrever um programa de computador. (STAIR, REYNOLDS, 2002). O *PHP: Hypertext Preprocessor* (PHP) é uma linguagem de

⁷ É uma plataforma de fonte aberta altamente modular para gerenciamento de conteúdo de *web*, com ênfase na colaboração. (TOMLINSON, 2012, p.1).

⁸ Tipo de *software* no qual os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o *software*. (CAMPOS, 2006).

código-aberto, de uso geral, utilizada especialmente para o desenvolvimento *web* e que pode ser embutida dentro do HTML. (ABT, 2017).

- Linguagem de marcação HTML (versão 5)

HTML é a sigla em inglês para *Hyper Text Markup Language*, ou, em português, linguagem para marcação de hipertexto. Através desta linguagem, programadores *web* criam conteúdos que podem ser interpretados pelos navegadores⁹. O conhecimento de HTML é o primeiro requisito para criar uma página *web*. (SILVA, 2016).

- Linguagem de estilo CSS (versão 3)

O CSS, abreviação de *Cascading Style Sheet* (Folha de Estilo em Cascata) é utilizado para definir todas as funções de apresentação de um documento desenvolvido com uma linguagem de marcação como o HTML, por exemplo. Enquanto o HTML é utilizado para definir a marcação e a estruturação de conteúdos, CSS delinea detalhes como cores de fontes, tamanhos de textos e todo o aspecto visual de um documento. (SILVA, 2016).

- Serviço de Banco de Dados MySQL (versão 5)

MySQL é um Sistema Gerenciador de Banco de Dados (SGBD) que utiliza a linguagem SQL¹⁰ (*Structure Query Language* – Linguagem de Consulta Estruturada). Um SGBD fornece uma *interface* entre o banco de dados e o usuário, como também entre o banco de dados e os programas. Através de um SGBD é possível criar, implementar, usar e atualizar um banco de dados. (STAIR, REYNOLDS, 2002).

- Servidor *web* Apache

Um servidor *web* consiste num programa de computador responsável por aceitar pedidos dos navegadores, e servi-los com respostas, como por exemplo, páginas *web* feitas em HTML, imagens, ou qualquer outro recurso que pode ser acessado pelo usuário (internauta). O Apache é um tipo de servidor *web* configurável, robusto, de alta performance, de fácil instalação e código fonte¹¹ aberto. (SOUZA, 2017).

- Drupal (versão 7)

Drupal é um sistema de gerenciamento de conteúdo *web*, com código fonte aberto, que permite a construção rápida e fácil de *web sites* simples ou complexos. Pode ser utilizado para diversas criações na *web* como blogs, portais de internet, páginas pessoais, *sites* de

⁹ Programa capaz de ler e apresentar ao usuário os conteúdos de um documento *web* escrito em linguagem de marcação. (SILVA, 2016, p.20).

¹⁰ Linguagem de manipulação de banco de dados. (STAIR, REYNOLDS, 2002, p.127).

¹¹ Termo que se refere a um *software* cujo código está disponível para *download* por qualquer pessoa e a uma filosofia de criação de programas voltada para a colaboração entre desenvolvedores. (KURTZ, 2016).

comércio eletrônico, jornais *on line*, redes sociais, *intranets*, dentre outros. (TOMLINSON, 2012).

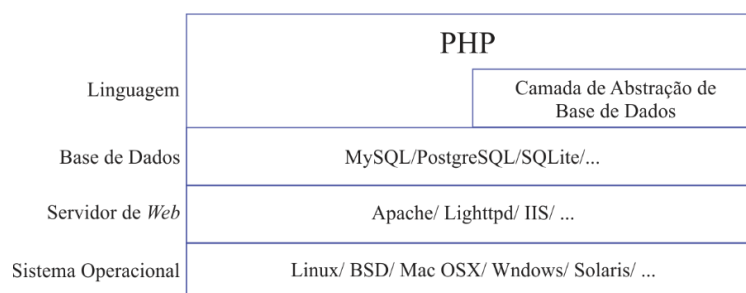
Um sistema gerenciador de conteúdo (*Content Management System_CMS*) facilita a inserção, alteração e exclusão de informações de *sites* sem a necessidade de utilizar de conhecimentos de programação. (LUIZ, 2014). É o que o Nosso Exercício oferece, por exemplo, ao permitir que qualquer usuário autenticado insira novos exercícios na plataforma. Neste caso, o usuário utiliza uma visão do sistema denominada *front-end*, ou seja, a visão do usuário final. (GABRIELI, 2006).

O Drupal é extensível, fornecido com um núcleo básico, porém, permite que novas funcionalidades sejam adicionadas através da ativação de módulos. Foi projetado para ser adaptado segundo as necessidades do usuário. (TOMLINSON, 2012). A modularidade é um dos princípios fundamentais do Drupal, tornando-o flexível e fazendo-o se destacar de outros gerenciadores de conteúdo.

Um *website* baseado em Drupal tem a sua funcionalidade aumentada através da habilitação de módulos já existentes, escritos por membros da comunidade Drupal ou ainda através da escrita de novos módulos por seus usuários. (TOMLINSON, 2012). A criação de novos módulos em Drupal exige um nível de conhecimento maior da tecnologia por parte do programador, já que o mesmo trabalha na visão *back-end* do sistema, criando regras de negócios e operações que são realizadas através do gerenciador de conteúdo. (LAMIM, 2014).

Drupal é escrito em PHP e se apoia em todas as tecnologias citadas anteriormente conforme é mostrado na Figura 10. Roda em qualquer sistema operacional e base de dados que suporte PHP. O Apache é o servidor *web* mais utilizado por Drupal, apesar de que outros também podem ser utilizados. O conhecimento e uso destas tecnologias são importantes quando há a necessidade de se criar novos módulos específicos que se adequem à realidade do *website* construído. (TOMLINSON, 2012).

Figura 10 - Tecnologias nas quais o Drupal se apoia



Fonte: TOMLINSON, 2012, p.2.

Além das tecnologias citadas para o desenvolvimento de novas funcionalidades para o *website* Nosso Exercício, outras ferramentas voltadas para o gerenciamento das atividades no projeto e controle de versões do código foram utilizadas.

- Trello

“O Trello é um sistema de quadro virtual para gerenciamento de tarefas que segue o método *Kanban*, muito usado no desenvolvimento com *Scrum*”. (SILVA, 2017). Este aplicativo pode ser utilizado para dar visibilidade às atividades executadas pelo Time *Scrum*, inclusive, durante o trabalho remoto realizado. Utilizando cartões virtuais é possível que cada membro do Time faça comentários para receber *feedback* imediato, carregue arquivos do seu computador ou da internet, adicione *checklists*, etiquetas, prazos e mais. (TRELLO, 2017).

- BitBucket

É um serviço gratuito de hospedagem para projetos de *software* que utilizam sistemas de controle de versão. Esse tipo de sistema auxilia equipes de *software* a gerenciar as alterações do código-fonte ao longo do tempo, acompanhando todas as modificações no código em um tipo especial de banco de dados. Manter um histórico completo de mudanças de longo prazo, permite voltar às versões anteriores do *software* para analisar as causas raiz dos erros encontrados, auxiliando na correção de problemas. (BITBUCKET, 2018).

4 METODOLOGIA

4.1 Classificação da pesquisa

Os conhecimentos gerados através desta pesquisa contribuíram na prática com o desenvolvimento de novas funcionalidades para o *website* Nosso Exercício, assim, este estudo pode ser classificado como de natureza aplicada. (PRODANOV; FREITAS, 2013). Além disso, classifica-se como intervencionista ao propor um *framework* capaz de auxiliar futuros desenvolvedores na evolução do *software* estudado e disponibilizar aos usuários da plataforma mais opções de apoio aos seus estudos e trabalhos acadêmicos. (VERGARA, 1990).

Quanto aos procedimentos técnicos utilizados, classifica-se como bibliográfica e documental. Bibliográfica porque toda a fundamentação teórica abordada originou-se de livros e publicações periódicas, impressas e eletrônicas. Documental, pois, também utilizou como fonte de pesquisa a documentação das linhas de código do *software* já existente, material sem tratamento analítico que sofreu modificações após a realização deste estudo. (GIL, 2002).

4.2 Pesquisa-Ação

Uma importante estratégia adotada para a realização deste trabalho foi a pesquisa-ação. Engel (2000, p. 183) menciona que “[...] a pesquisa-ação pode ser aplicada em qualquer ambiente de interação social que se caracterize por um problema, no qual estão envolvidos pessoas, tarefas e procedimentos”. Assim, este tipo de pesquisa adequa-se muito bem a projetos *Scrum* em que pessoas trabalham juntas para atingirem um objetivo comum, executando tarefas segundo os princípios deste método ágil.

Thiollent define a pesquisa-ação como:

[...] um tipo de pesquisa com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo e no qual os pesquisadores e participantes representativos da situação ou do problema estão envolvidos de modo cooperativo ou participativo. (THIOLLENT, 2011, p. 20).

A implantação do *Scrum* na realidade do projeto Nosso Exercício envolveu a participação de uma equipe engajada (incluindo esta pesquisadora), que, colaborativamente, realizou as atividades de desenvolvimento do *software web* proposto. Os resultados alcançados em cada etapa do projeto dependeram, dentre vários fatores, do nível de comprometimento desta equipe.

A pesquisa-ação é considerada uma estratégia metodológica da pesquisa social.

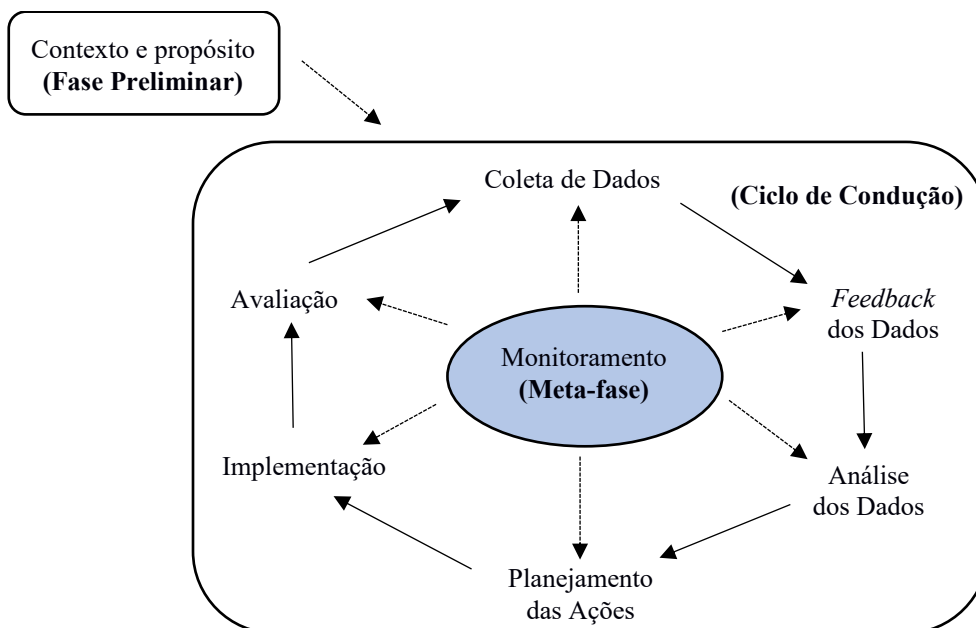
Alguns dos seus principais aspectos resumidos por Thiollent são:

- Há uma ampla e explícita interação entre pesquisadores e pessoas implicadas na situação investigada.
- Desta interação resulta a ordem de prioridade dos problemas a serem pesquisados e das soluções a serem encaminhadas sob forma de ação correta.
- O objetivo de investigação não é constituído pelas pessoas e sim pela situação social e pelos problemas de diferentes naturezas encontradas nesta situação.
- O objetivo da pesquisa ação consiste em resolver, ou pelo menos, em esclarecer, os problemas da situação observada.
- Há, durante o processo, um acompanhamento das decisões, das ações e de toda a atividade intencional dos atores da situação.
- A pesquisa não se limita a uma forma de ação (risco de ativismo): pretende-se aumentar o “conhecimento dos pesquisadores” e o conhecimento ou nível de consciência das pessoas ou grupos considerados. (THIOLLENT, 2011, p. 22-23).

O autor menciona ainda a flexibilidade do planejamento de uma pesquisa-ação em que as ações são realizadas pelos próprios atores sociais, podendo ser o pesquisador um animador ou um participante ativo. Através deste tipo de pesquisa é possível estudar ativamente os problemas, decisões, ações, negociações, conflitos e conscientizações que ocorrem com os agentes durante o processo, gerando um ganho de conhecimento sobre o objeto de estudo.

Apresenta-se na Figura 11 o ciclo de uma pesquisa-ação através de um modelo sugerido por Coughlan e Coughlan (2002).

Figura 11 - Ciclo da pesquisa-ação



Fonte: COUGHLAN E COUGHLAN, 2002, p. 230.

O modelo compreende três fases: fase preliminar (ou pré-fase), ciclo de condução (ou desenvolvimento) e meta-fase (ou monitoramento). Segundo os autores, na fase preliminar o contexto onde a pesquisa será realizada e o seu objetivo são compreendidos. Identifica-se as partes interessadas no projeto, as suas expectativas e visão sobre o estado atual deste. Além disso, as justificativas para o desenvolvimento das ações e da pesquisa são definidas.

O ciclo de condução ou desenvolvimento é realizado em seis etapas: coleta dos dados, *feedback* dos dados, análise dos dados, planejamento da ação, implementação da ação e avaliação dos dados, reiniciando uma nova coleta dos dados quando for necessária, finalizando assim, o ciclo. Estes ciclos podem se repetir até que se atinja o objetivo almejado pela pesquisa. No Quadro 3, é apresentada uma síntese desses seis passos conforme o modelo proposto por Coughlan e Coughlan (2002).

Quadro 3- Etapas do Ciclo de Condução

Fase	Descrição	Meios
Coleta de Dados	Dados são gerados por meio do envolvimento com o processo organizacional.	Dados qualitativos: observação direta, discussões, entrevistas Dados quantitativos: relatórios, registros operacionais.
<i>Feedback</i> dos Dados	Os dados são retornados para a organização visando disponibilizá-los para análise.	Relatórios elaborados pelo pesquisador, reuniões de <i>feedback</i> .
Análise dos Dados	Análise conjunta realizada pelo pesquisador e membros envolvidos (por exemplo, membros dos times de trabalho).	Ferramentas e critérios de análise que necessitam estar relacionados aos propósitos da pesquisa e da intervenção.
Planejamento da Ação	Atividade conjunta que estabelece o que vai ser feito e com que prazo.	Responder questões do tipo: o que necessita ser alterado e em que parte da organização? Qual o apoio necessário? Como o comprometimento pode ser obtido? Como superar as resistências?
Implementação da Ação	A ação estabelecida é então implementada visando promover as mudanças planejadas.	Ferramentas estabelecidas para executar a implantação em colaboração com os envolvidos.
Avaliação	Reflexão dos resultados esperados ou não, decorrentes da implementação da ação	Revisão do processo visando avaliar os resultados, incluindo melhorias para o ciclo seguinte.

Fonte: MIGUEL, 2011, p. 64.

Seguindo ainda a ideia de Coughlan e Coughlan (2002), durante a meta-fase ou monitoramento, faz-se a verificação de cada etapa realizada no ciclo de desenvolvimento buscando identificar qual foi o aprendizado gerado na pesquisa-ação, mostrando assim os resultados alcançados. Com intuito de descrever a dinâmica conduzida durante este trabalho, o ciclo da pesquisa-ação proposto por estes autores, mais especificamente o ciclo de condução,

foi associado ao ciclo de desenvolvimento dos *Sprints* no *Scrum*. O Quadro 4 demonstra um esquema desta associação.

Quadro 4 - Associação do ciclo da pesquisa-ação ao ciclo de desenvolvimento dos *Sprints* no *Scrum*

Fase Preliminar	
Ciclo de Condução	
<i>Sprints</i>	Desenvolvimento
<i>Sprint Planning</i> do <i>Sprint</i>	Coleta, <i>Feedback</i> dos dados, Análise dos dados e Planejamento da ação
Execução do <i>Sprint</i>	Implementação da ação
<i>Sprint Review</i> e Retrospectiva	Avaliação dos dados
Meta-fase (Monitoramento)	

Fonte: Própria autora.

O quadro demonstra que, semelhante à pesquisa-ação, o ponto de partida de um projeto *Scrum* é a definição do contexto em que o mesmo será aplicado (fase preliminar). Feito isto, os *Sprints* são inicializados (ciclo de condução). Durante as Reuniões de Planejamento dos *Sprints* os dados são coletados junto ao *Product Owner*, que, através do seu *feedback* orienta o Time sobre as suas reais necessidades em relação ao *software* que será construído. Assim, a equipe analisa detalhadamente os dados levantados e realiza o planejamento das ações que serão executadas durante o *Sprint*. Após a implementação destas ações (Execução do *Sprint*), ocorrem as Reuniões de Revisão e Retrospectiva para que os resultados obtidos sejam avaliados. Por fim, os resultados de cada *Sprint* são analisados, aprimorando o processo através do aprendizado gerado (meta-fase). Detalhes sobre como este esquema foi utilizado neste trabalho serão descritos nas próximas seções.

A implantação do *Scrum* no Nosso Exercício ocorreu em duas etapas e envolveu duas formações para o Time. A primeira etapa, objetivou realizar uma capacitação sobre *Scrum* e as ferramentas tecnológicas utilizadas no desenvolvimento do Nosso Exercício. Já a segunda, tratou do desenvolvimento das funcionalidades para este *website*.

Quadro 5 - Esquema básico da pesquisa ação aplicada neste projeto

Pesquisa Ação no Projeto Nosso Exercício	
Fase Preliminar- Contexto Geral da Pesquisa	
Etapa 1	Fase Preliminar da Etapa 1
	Ciclo de Condução da Etapa 1
	Meta-fase da Etapa 1
Etapa 2	Fase Preliminar da Etapa 2
	Ciclo de Condução da Etapa 2
	Meta-fase da Etapa 2

Fonte: Própria autora

O Quadro 5 apresenta um esquema de como a pesquisa ação foi abordada neste trabalho. Inicialmente, apresenta-se na ‘Fase Preliminar-Contexto geral’ o cenário onde a pesquisa foi realizada, identificando a situação-problema e o seu propósito. Em seguida, apresenta-se as etapas, cada uma com as atividades básicas realizadas durante a pesquisa-ação, a saber, a fase preliminar, o ciclo de condução e a meta-fase.

5 DESCRIÇÃO DA PESQUISA-AÇÃO

5.1 Fase Preliminar – Contexto geral da pesquisa

Esta fase visou a compreensão mais detalhada do *software web* a ser desenvolvido e a problemática envolvida na sua evolução. Apresenta-se as partes interessadas no projeto bem como a equipe formada para a sua execução. Ademais, confirma-se o comprometimento dos participantes e interessados neste projeto.

O desenvolvimento do Nosso Exercício iniciou no segundo semestre de 2015 e sua primeira versão foi lançada no primeiro semestre de 2016. Todas as funcionalidades disponíveis até então foram de responsabilidade de apenas um programador, professor do curso de Matemática da UFVJM - Campus Mucuri. Apesar do bom trabalho realizado, as atividades de desenvolvimento do *software web* executadas por este profissional, resumiram-se na programação, desconsiderando a aplicação formal dos princípios básicos da Engenharia de *Software*. Assim, não houve nenhum registro do que foi feito e como foi feito (exceto pelas linhas de código); nenhum estudo dos requisitos desejáveis para o *software* a curto, médio ou longo prazo; nenhum planejamento realizado ou seguido, dificultando assim a sua manutenção e evolução por outras pessoas.

Tudo isto poderia não ser um problema caso o Nosso Exercício não precisasse evoluir. Buscando auxílio para acrescentar novas funcionalidades à plataforma, este professor iniciou uma parceria com o PET, passando a contar a partir daí, com a colaboração de bolsistas deste programa. A ideia inicial era de que estes alunos, graduandos do Bacharelado em Ciência e Tecnologia (BC&T) e do curso de Matemática da UFVJM - Campus Mucuri, dessem continuidade ao projeto, assumindo a função de novos programadores.

Apesar de não terem formação na área da Computação, os alunos já haviam cursado duas disciplinas de programação básica na Instituição. Assim surgiu a equipe do projeto Nosso Exercício, constituída pelo professor programador como coordenador, pelos alunos bolsistas e por outro professor coordenador do PET, como colaboradores.

Acreditando que os alunos envolvidos tivessem base suficiente para iniciar na área de desenvolvimento de *software*, foi adquirido um livro sobre Desenvolvimento em Drupal 7, ambiente no qual o Nosso Exercício foi desenvolvido, para que os alunos iniciassem estudos em grupo e fossem se familiarizando com o CMS. Mesmo com o suporte oferecido pelo professor coordenador no esclarecimento das dúvidas sobre a tecnologia proposta, percebeu-se

que a pequena experiência dos alunos nesta área do conhecimento, tornou-se um obstáculo para a continuidade do projeto. Constatou-se que haveria necessidade de treinamentos e acompanhamentos mais intensivos.

A inserção de uma nova integrante à equipe do Nosso Exercício, aluna do mestrado em Tecnologia Ambiente e Sociedade (TAS) da UFVJM-Campus Mucuri, com formação em Ciência da Computação (esta pesquisadora), permitiu o seguimento do projeto. Após inúmeras conversas informais com os professores responsáveis, ficou decidido que a nova colaboradora assumiria o Nosso Exercício como seu projeto de curso.

Embasada na sua experiência docente há 10 anos ministrando a disciplina de Engenharia de *Software*, atuando em projetos didáticos sobre metodologias ágeis, este novo membro propôs o uso do *Framework Scrum* para gerenciar o processo de desenvolvimento das novas funcionalidades que deveriam ser incorporadas ao *website* Nosso Exercício. A escolha do *Scrum* para este projeto, além das características já mencionadas anteriormente, deu-se em virtude de alguns fatores, a saber:

- Da flexibilidade e simplicidade do método, permitindo uma adequação à realidade da equipe disponível e facilidade na sua compreensão.

- Do ciclo de vida iterativo e incremental, favorecendo entregas frequentes aos clientes e um aprendizado contínuo da equipe envolvida.

- Da característica do tipo de *software* a ser desenvolvido. Métodos ágeis adaptam-se facilmente à dinâmica evolutiva das aplicações *web*.

- Da transparência do método que permite aos seus idealizadores uma visão realista do andamento do projeto durante todo o tempo.

Apresentadas todas estas características aos responsáveis pelo projeto e com o consentimento destes, firmou-se um acordo sobre o uso do *framework* proposto. Neste momento, um novo colaborador, externo à Instituição, com experiência prática em desenvolvimento de *software*, passou a fazer parte da equipe. A partir daí o Time *Scrum* foi formado e novos planos foram traçados para o prosseguimento do projeto.

A primeira composição do Time *Scrum* com a definição dos papéis assumidos por cada membro pode ser conferida no Quadro 6.

Quadro 6 - Formação da primeira equipe de trabalho *Scrum* no Nosso Exercício

Papéis	Membros
<i>Product Owner</i>	Professor coordenador
<i>Scrum Master</i>	Pesquisadora (aluna do mestrado)
Time de Desenvolvimento	4 membros do PET e 1 membro externo colaborador

Fonte: Própria autora.

Com intuito de se obter um entendimento mais aprofundado do *software* a ser desenvolvido, foram realizadas reuniões com o professor coordenador. Através destas, foi possível fazer o levantamento e registro das funcionalidades (requisitos) existentes e desejáveis do *website*. O Quadro 7 demonstra o resultado do levantamento realizado.

Quadro 7 - Funcionalidades existentes e desejáveis para o *website* Nosso Exercício

Funcionalidades Existentes	Funcionalidades Desejáveis
- Pesquisar exercícios	- Permitir o <i>download</i> das listas de exercícios.
- Gerenciar lista (criar, editar, excluir)	- Permitir que o usuário crie uma lista com uma certa quantidade 'n' de exercícios escolhidos aleatoriamente de certo conteúdo;
- Gerenciar exercícios (adicionar, editar, excluir)	- Permitir criação de questões do tipo múltipla escolha.
- Inserir comentários nos exercícios	- Permitir que o usuário crie grupos, de modo que apenas aquele grupo veja uma certa lista.
- Avaliar nível de dificuldade dos exercícios	- Gravar os exercícios sendo digitados como 'Rascunho', de modo que se algum erro de conexão ocorrer o usuário não perca o seu trabalho.
- Denunciar erro (nos casos de erros nas soluções dos exercícios)	- Em listas públicas, permitir para o usuário anônimo, exibir <i>link</i> para visualização completa do exercício (atualmente isto só é possível com usuários autenticados).
- Denunciar infração (nos casos em que haja infração dos direitos autorais)	- Ter um conjunto de <i>links</i> ao lado da barra de busca que exiba todos os exercícios por uma certa disciplina.
	- Melhorar o <i>layout</i> /usabilidade da plataforma.
	- Realizar buscas apenas por exercícios com comentários e/ou respostas .
	- Ao realizar uma pesquisa, retornar a quantidade de itens resultantes da busca.
	- Ao compartilhar a lista com alguém, permitir a opção 'com' ou 'sem resposta'.

Fonte: Própria autora.

As funcionalidades desejáveis foram transformadas posteriormente no *Product Backlog* do produto, lista ordenada e priorizada conforme solicitação do *Product Owner*. Importante esclarecer que esta lista é dinâmica e pode ser "atualizada, reordenada e refinada de acordo com o nível de detalhes que é possível de se ter em cada momento do projeto". (SABBAGH, 2016, p. 148). O Quadro 8 mostra a primeira versão criada para o *Product Backlog* onde cada requisito recebeu um identificador e foi classificado como "Melhoria" ou "Nova Funcionalidade".

Quadro 8 - Primeira versão criada do *Product Backlog*

<i>Product Backlog - Projeto Nosso Exercício</i>		
ID	Descrição	Classificação
1	Permitir o <i>download</i> de listas de exercícios nos formatos .doc (<i>Word</i>), .odt (<i>LibreOffice</i>), .pdf (<i>Acrobat</i>) e .tex (<i>LaTeX</i>);	Nova Funcionalidade
2	Permitir que o usuário crie uma lista com uma certa quantidade 'n' de exercícios escolhidos aleatoriamente de certo conteúdo;	Nova Funcionalidade
3	Permitir criação de questões do tipo múltipla escolha onde o usuário possa selecionar uma alternativa e a plataforma guardar se ele acertou ou errou a questão.	Nova Funcionalidade
4	Permitir que o usuário crie grupos, de modo que apenas aquele grupo verá uma certa lista.	Nova Funcionalidade
5	Gravar os exercícios sendo digitados como 'Rascunho', de modo que se algum erro de conexão ocorrer o usuário não perca o seu trabalho.	Nova Funcionalidade
6	Em listas públicas, permitir para o usuário anônimo, exibir <i>link</i> para visualização completa do exercício (atualmente isto só é possível com usuários autenticados).	Melhoria
7	Ter um conjunto de <i>links</i> ao lado da barra de busca que exiba todos os exercícios por uma certa disciplina.	Melhoria
8	Melhorar o <i>layout</i> /usabilidade da plataforma.	Melhoria
9	Realizar buscas apenas por exercícios com comentários e/ou respostas .	Melhoria
10	Ao realizar uma pesquisa, retornar a quantidade de itens resultantes da busca (Ex: 10 itens).	Melhoria
11	Ao compartilhar a lista com alguém, permitir a opção 'com' ou 'sem resposta'. Pensar em professores que queiram compartilhar a lista com seus alunos, mas não queiram disponibilizar as respostas naquele momento.	Melhoria

Fonte: Própria autora.

Além disso, durante as reuniões buscou-se informações sobre os perfis de usuários existentes na plataforma e suas principais permissões no *website* conforme mostrado no Quadro 9.

Quadro 9 - Usuários do Nosso Exercício e suas permissões antes deste projeto

Permissões	Perfis de Usuários			
	Anônimo	Autenticado	Editor	Administrador
- Pesquisar exercícios	X	X	X	X
- Criar e editar seus próprios exercícios e/ou listas de exercícios		X	X	X
- Apagar seus próprios exercícios e/ou listas de exercícios			X	X
- Editar e apagar exercícios e/ou listas de exercícios de outros usuários.			X	X
- Avaliar nível de dificuldade dos exercícios	X	X	X	X
- Visualizar comentários dos exercícios	X	X	X	X
- Inserir comentários nos exercícios		X	X	X
- Denunciar erro e/ou infração	X	X	X	X
- Alterar configurações avançadas (painel administrativo, código, <i>layout</i> , permissões de usuários, etc)				X

Fonte: Própria autora

Com o Time *Scrum* formado, os requisitos levantados e a realidade do *software* conhecida, o próximo passo foi investir na qualificação da equipe através da realização de dois treinamentos: um voltado para as tecnologias de desenvolvimento do *software web* e outro sobre o *framework Scrum*.

O ‘Curso Básico de Desenvolvimento Web’, foi ministrado pelo professor coordenador, no período de 04/07/2016 a 06/09/2016, no laboratório de Informática da UFVJM-Campus Mucuri, com carga horária de 40 horas. O conteúdo trabalhado abordou a linguagem de programação PHP, a linguagem de marcação HTML, a linguagem de estilo CSS e o banco de dados MYSQL.

No dia 14 de dezembro de 2016, foi realizado um minicurso de 5 horas na mesma Instituição, sobre o *Scrum*, ministrado pela aluna do mestrado, para que o Time pudesse conhecer o *framework* e suas principais características.

5.2 Etapa 1: Uso do *Scrum* na capacitação de alunos em desenvolvimento de *software*

5.2.1 Fase Preliminar da Etapa 1

A Etapa 1 foi realizada no período de 15/03/2017 a 28/04/2017. Alguns riscos já eram conhecidos antes mesmo do início do trabalho, como por exemplo, o nível heterogêneo de conhecimento e experiência nas áreas do saber do Time *Scrum* formado, necessárias para a condução do projeto, conforme indicado no Quadro 10.

Quadro 10 - Nível de conhecimento/experiência de cada papel nas áreas do saber necessárias ao projeto

Papéis	Nível de conhecimento e experiência			
	PHP, HTML, CSS, MySQL	Drupal	<i>Scrum</i>	Nosso Exercício
<i>Scrum Master</i>	Médio	Baixo	Alto	Médio
Time (Alunos do PET)	Baixo	Baixo	Baixo	Alto
Time (Colaborador Externo)	Alto	Baixo	Médio	Baixo

Fonte: Própria autora.

Sabbagh (2016, p. 97) menciona que apesar de multidisciplinar, um Time *Scrum* não possui, necessariamente, membros multidisciplinares. O autor reconhece que as pessoas possuem suas especialidades, conhecimentos e áreas de interesse específicos. Como o *Scrum* estimula os membros do Time a trabalharem juntos, acabam trocando conhecimentos e desenvolvendo habilidades secundárias. Assim, ao passar do tempo, o Time tende a nivelar os seus conhecimentos e habilidades. Isto reduz o risco de não se cumprir a Meta do *Sprint* e

também as chances de que, em alguns momentos, algum membro do Time fique sem ter o que fazer.

Ao discutir e analisar os itens do *Product Backlog* com o Time, mesmo com os treinamentos realizados, foi identificada uma insegurança elevada de alguns membros (alunos do PET) com as tecnologias envolvidas no desenvolvimento do *website* e até mesmo com a prática do *Scrum*. A fim de dirimir estas deficiências, ficou decidido que mais uma capacitação seria realizada, agora focada na prática, considerando o contexto do Nosso Exercício.

Esta etapa focou nesta capacitação que visou utilizar o *Scrum* para treinamento do próprio *Scrum*, incluindo também as tecnologias *web* envolvidas (PHP, HTML, CSS e MYSQL). Importante relatar que a qualificação dos alunos bolsistas consiste em um dos objetivos do projeto Nosso Exercício, conforme descrito em sua documentação de registro. Pretendia-se que, através desta experiência, os alunos alcançassem um nível de conhecimento que lhes permitissem desenvolver as funcionalidades solicitadas para a evolução da plataforma.

O Drupal, ambiente no qual o Nosso Exercício havia sido desenvolvido, não foi contemplado nesta capacitação, apenas as tecnologias que o apoiam, pré-requisitos para seu aprendizado. Isto porque, até então, nenhum membro do Time de Desenvolvimento havia tido algum tipo de experiência válida com o mesmo. A princípio, isto não foi visto como um empecilho, pois, a *Scrum Master* e o membro colaborador externo já estavam se empenhando neste aprendizado. Assim, os itens do *Product Backlog* foram adaptados dentro do contexto do Nosso Exercício, ao nível do conhecimento prévio do Time em relação às tecnologias mencionadas.

Seguindo a proposta de adaptação do ciclo da pesquisa-ação ao ciclo de vida do *Scrum*, na próxima seção descreve-se o Ciclo de Condução (desenvolvimento) da Etapa 1 deste projeto.

5.2.2 Ciclo de Condução: Etapa 1

O primeiro dia desta etapa foi dedicado para o planejamento do projeto. A *Scrum Master* apresentou ao Time o *Product Backlog* e os papéis que cada membro deveria assumir. Foi reforçado pela *Scrum Master* as ações esperadas de cada papel e a importância do comprometimento no trabalho realizado. Devido à mudança repentina de cidade do *Product Owner*, a *Scrum Master* ficou responsável por assumir também o seu papel e ser a porta-voz do

andamento das atividades para este membro, que manteve a sua participação no projeto através de videoconferência.

Um cronograma foi definido considerando as disponibilidades dos componentes do grupo, já que os mesmos se ocupavam com outras atividades além do projeto. Ficou determinado que o Time se reuniria presencialmente todas as quartas-feiras, das 16h às 18h e que o trabalho seria prosseguido remotamente com mais 4 horas de dedicação, totalizando 6 horas semanais. De fato, Brod (2015) recomenda que mesmo que o Time trabalhe, na sua maior parte, remotamente, é importante estabelecer momentos em que todos possam se encontrar em um mesmo local. A *Scrum Master* e o colaborador externo, membros mais experientes, se dispuseram a dar a assistência necessária ao Time de Desenvolvimento em caso de dúvidas.

O primeiro planejamento traçado para o projeto, com os novos requisitos adaptados dentro do contexto do Nosso Exercício e do nível de conhecimento apresentado pelos alunos, com a quantidade prevista e duração dos *Sprints*, pode ser conferido no Quadro 11. Conforme pode-se observar no quadro, o projeto foi dividido em quatro *Sprints*, cada um com duração de uma semana, exceto o último, pois, o planejamento teve que se adequar ao calendário letivo, finalizando esta etapa do projeto juntamente com o período de aulas. Assim, as funcionalidades distribuídas aos membros do Time foram adequadas ao tempo disponível para o seu desenvolvimento.

Quadro 11 - Planejamento Inicial (Etapa 1)

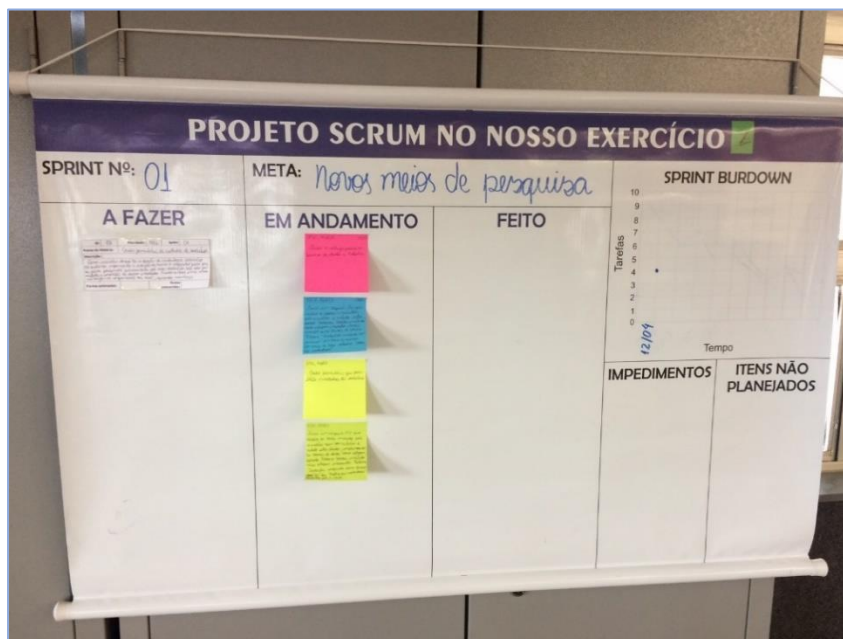
Data	Sprint	Eventos	Requisitos
15/03/17		<i>Sprint Planning</i> do Projeto (2h)	
22/03/17	1	<i>Sprint Planning</i> do <i>Sprint 1</i> (1h). Execução do <i>Sprint 1</i> (1h_ Desenvolvimento/Suporte presencial). (Mais 4 horas remotamente)	Permitir o cadastro de exercícios didáticos.
29/03/17	1 e 2	Reuniões de Revisão e Retrospectiva do <i>Sprint 1</i> (1h). <i>Sprint Planning</i> do <i>Sprint 2</i> (1h). (Mais 4 horas remotamente)	Permitir a consulta dos exercícios didáticos cadastrados.
05/04/17	2	Execução da <i>Sprint 2</i> (1h_ Desenvolvimento/Suporte presencial). Reuniões de Revisão e Retrospectiva da <i>Sprint 2</i> (1h).	
12/04/17	3	<i>Sprint Planning</i> da <i>Sprint 3</i> (1h). Execução da <i>Sprint 3</i> (1h_ Desenvolvimento/Suporte presencial). (Mais 4 horas remotamente)	Permitir a criação de listas de exercícios pelos usuários.
19/04/17	3 e 4	Reuniões de Revisão e Retrospectiva da <i>Sprint 3</i> (1h). <i>Sprint Planning</i> da <i>Sprint 4</i> (1h). (Mais 4 horas remotamente)	Definir o <i>layout</i> e realizar a integração do <i>software</i> .
26/04/17	4	Execução da <i>Sprint 4</i> (1h_ Desenvolvimento/Suporte presencial). Reuniões de Revisão e Retrospectiva da <i>Sprint 4</i> (1h).	
28/04/17		Apresentação do <i>software</i> .	

Fonte: Própria autora.

O conceito de ‘Pronto’ foi apresentado ao Time. Nesse projeto uma tarefa pronta significava que a funcionalidade havia sido desenvolvida, testada individualmente pelo membro que a fez, testada após a integração com as demais tarefas e, por fim, testada e aprovada pela *Scrum Master*.

Foi providenciado para acompanhamento das tarefas um quadro *Kanban*, que ficou exposto no ambiente em que as reuniões presenciais aconteceram, visível a todos os integrantes. Neste quadro, as histórias de usuário de cada *Sprint* foram anexadas na coluna “A Fazer” e divididas em tarefas dispostas em *post-its* (notas adesivas), que mudavam de posição no quadro toda vez que um membro prosseguia nas suas atividades. Os *post-its* foram separados por cor, na qual, cada uma representava um membro do Time de Desenvolvimento (Figura 12). Assim, era facilmente possível identificar a tarefa de cada membro e o *status* em que se encontrava.

Figura 12 - Quadro *Kanban* utilizado na Etapa 1 do projeto



Fonte: Própria autora.

A Figura 13 demonstra um exemplo de história de usuário, anexada ao quadro *Kanban*, usada para descrever uma funcionalidade.

Figura 13 - Exemplo de história de usuário utilizada no projeto

SPRINT Nº: 02

A FAZER

ID:	Prioridade: Alta	Sprint: 02
Nome da História:	Criar as consultas das execuções	
Descrição:	Como usuário desejo realizar consultas das execuções (por disciplina e tópicos) para resolvê-los ou organizá-los em uma lista.	
Pontos estimados:		Pontos consumidos:

Fonte: Própria autora.

Como o Time em determinados momentos executava o trabalho de desenvolvimento remotamente, para que não se perdesse a visualização do que estava sendo realizado, foi criado também um *Kanban* digital. Para isso foi utilizada a ferramenta Trello. Através deste quadro, todos os integrantes da equipe *Scrum*, incluindo o *Product Owner*, a qualquer momento podiam ter a visualização do andamento do projeto.

Da mesma forma que no quadro físico, cada cartão adicionado no Trello correspondia a uma tarefa que deveria ser executada por um dos membros do Time (indicados por suas iniciais no canto inferior direito de cada cartão, conforme destacado em vermelho na Figura 14). Cada membro movia o seu cartão à medida que o seu trabalho avançava. A Figura 14 demonstra uma tela do Trello no final de um *Sprint*.

Figura 14 - Quadro *Kanban* Digital



Fonte: Própria autora.

O *Sprint* 1 seria iniciado no dia 22/03/17. Problemas de adequação da rotina de alguns membros do Time fizeram com que o primeiro planejamento feito, tivesse de ser modificado. Como o projeto possuía uma data de término bem definida, o planejamento precisou ser adaptado. Assim, o tempo proposto para o desenvolvimento/suporte durante os encontros presenciais foi destinado, em sua maior parte para as reuniões de Revisão e Retrospectiva do *Sprint* realizado e reuniões de Planejamento do novo *Sprint* a ser considerado. O novo planejamento feito pode ser conferido no Quadro 12.

Quadro 12 - Segundo Planejamento (Etapa 1)

Data	Sprint	Eventos	Requisitos
15/03/17		<i>Sprint Planning</i> do Projeto (2h)	
22/03/17		Cancelado	
29/03/17	1	<i>Sprint Planning</i> do <i>Sprint</i> 1 (1h). Execução do <i>Sprint</i> 1 (1h_Desenvolvimento/Suporte presencial). (Mais 4 horas remotamente)	Permitir o cadastro de exercícios didáticos.
05/04/17	1 e 2	Reuniões de Revisão e Retrospectiva do <i>Sprint</i> 1 (1h). <i>Sprint Planning</i> do <i>Sprint</i> 2 (1h). (Mais 4 horas remotamente)	Permitir a consulta dos exercícios didáticos cadastrados.
12/04/17	2 e 3	Reuniões de Revisão e Retrospectiva da <i>Sprint</i> 2 (1h). <i>Sprint Planning</i> da <i>Sprint</i> 3 (1h).	Permitir a criação de listas de exercícios pelos usuários.
19/04/17	3 e 4	Reuniões de Revisão e Retrospectiva da <i>Sprint</i> 3 (1h). <i>Sprint Planning</i> do <i>Sprint</i> 4 (1h). (Mais 4 horas remotamente)	Definir o <i>layout</i> e realizar a integração do <i>software</i> .
26/04/17	4	Reuniões de Revisão e Retrospectiva da <i>Sprint</i> 4 (1h). Suporte para integração e apresentação final do <i>software</i> .	
28/04/17		Apresentação do <i>software</i> .	

Fonte: Própria autora.

O *Sprint* 1 iniciou no dia 29/03/17. A princípio realizou-se a *Sprint Planning* onde a *Scrum Master* apresentou a história de usuário para este *Sprint* (Quadro 13). Esta história foi dividida em 4 tarefas que deveriam ser desenvolvidas por cada membro do Time. Tanto a história como as tarefas foram afixadas no quadro *Kanban* físico. Os membros foram orientados pela *Scrum Master* sobre como deveriam mover as tarefas no Trello (*Kanban* digital) e a importância deste quadro para a visibilidade do projeto.

Quadro 13 - História de Usuário do *Sprint 1*

ID:	1	Prioridade:	Alta	<i>Sprint:</i>	1
Nome da História:	Criar formulário de cadastro dos exercícios				
Descrição:	<p>Como usuário desejo ter a opção de cadastrar exercícios no sistema, informando a disciplina, termo e descrição para que eu possa futuramente pesquisar por estes exercícios.</p>				
Pontos estimados:		Pontos consumidos:			

Fonte: Própria autora.

Durante a *Sprint Planning* o Time discutiu sobre as tarefas a serem realizadas, buscando nivelar o entendimento sobre as mesmas. A *Scrum Master* e o membro colaborador externo auxiliaram o Time no esclarecimento do que deveria ser feito. Devido a inexperiência do Time, o tempo dedicado a esta reunião (2 horas) foi maior do que o previsto (1 hora), pois, várias dúvidas relacionadas ao desenvolvimento surgiram e buscaram ser sanadas para que a equipe tivesse condição de realizar as atividades propostas. Assim, nenhum trabalho de codificação ocorreu neste dia conforme planejado. Ficou decidido que, após as orientações dadas, o Time tentaria desenvolver as atividades deste *Sprint* remotamente e poderia contar com o suporte da *Scrum Master* e do membro colaborador externo.

Através do Trello foi possível acompanhar em qual história cada membro estava trabalhando, se estava dentro do prazo ou atrasado, e, se algum impedimento estava acontecendo. Durante o projeto foi proposto que o quadro também fosse utilizado para registrar o tempo de desenvolvimento de cada tarefa realizada pelo Time, bem como para postar o código criado e/ou qualquer outra observação que se fizesse necessária registrar.

No dia 05/04/17 foi dada a continuidade do *Sprint 1*. A *Scrum Master* orientou o Time sobre como o dia deveria ser conduzido. Inicialmente, o Time atualizou o quadro *Kanban* físico. O Trello já havia sido atualizado e indicava que todos os membros haviam cumprido com as suas tarefas. O resultado do trabalho deste *Sprint* pode ser conferido na Figura 15.

Figura 15- Tela de cadastro de exercícios (História 1- Etapa 1)

Cadastro Exercício

Disciplina*: Matemática

Termo*: Funções

Descricao*:
 O gráfico da função $f(x) = mx + n$ passa pelos pontos $(-1, 3)$ e $(2, 7)$. Calcule o valor de m .

Resolucao
 Caso tenha a resolução

* = Preenchimento obrigatório

Cadastrar

Fonte: Própria autora.

Durante a *Sprint Review* cada membro apresentou ao Time o resultado do seu trabalho e o raciocínio utilizado para o mesmo, compartilhando assim o conhecimento entre o grupo. Das quatro tarefas apresentadas, duas atenderam ao que foi pedido, e duas não. Dentre estas duas que não atenderam, uma apresentou erro e outra apenas não foi testada, porém, o membro avançou a tarefa nos quadros de andamento como se tivesse finalizada. Nesta oportunidade a *Scrum Master* lembrou ao Time o conceito de ‘Pronto’.

Ainda nesta reunião, discutiu-se sobre a necessidade de adoção de uma padronização do código-fonte¹² gerado pela equipe, uma vez que a intenção seria integrar ao longo do projeto todos os códigos gerados para a criação de uma aplicação única. A padronização facilitaria a leitura e entendimento dos membros do Time quando precisassem usar qualquer parte do código gerado.

A *Retrospective Review* aconteceu logo em seguida. Neste momento a equipe teve a oportunidade de rever o que deu certo e poderia ser mantido, e o que não deu e deveria ser corrigido. O esforço do Time em tentar realizar as suas tarefas, demonstrando comprometimento, o uso adequado do Trello e o compartilhamento de como as tarefas foram realizadas, explicitadas na *Sprint Review*, foram alguns dos itens apontados como positivos neste *Sprint*. O atraso de alguns membros, a falta de padronização no código, o não atendimento ao conceito de ‘Pronto’ e o curto tempo disponível para os encontros presenciais, foram os pontos negativos levantados pelo grupo. A falta de registro do tempo consumido para o desenvolvimento de cada tarefa também foi outro ponto negativo evidenciado, pois, sem este registro não seria possível fazer uma estimativa para as funcionalidades futuras.

¹² Sistema de símbolos utilizado para codificar um programa de computador. (DICIONÁRIO INFOPÉDIA DA LÍNGUA PORTUGUESA, 2017).

Ainda no dia 05/04/17 foi realizada a *Sprint Planning* do *Sprint 2*. A história deste *Sprint* (Quadro 14) foi apresentada ao Time de Desenvolvimento pela *Scrum Master*, bem como as suas tarefas. Após um longo debate sobre os detalhes de cada tarefa, percebeu-se que o Time estava inseguro para desenvolvê-las, pois exigiam conhecimento de banco de dados. Apesar de terem tido no treinamento do curso de ‘Desenvolvimento *Web*’ uma parte do tempo voltada para este conteúdo, a equipe demonstrou não se sentir pronta para adotá-lo no próximo *Sprint*.

Quadro 14 - História de Usuário do *Sprint 2*

ID:	2	Prioridade:	Alta	<i>Sprint</i>:	2
Nome da História:	Gerar consultas dos exercícios				
Descrição:	<p>Como usuário desejo pesquisar os exercícios cadastrados no sistema, inserindo em um campo de busca o nome de uma disciplina ou um termo qualquer que contenha na descrição de um exercício, para resolvê-los ou organizá-los em uma lista.</p>				
Pontos estimados:		Pontos consumidos:			

Fonte: Própria autora.

Ficou acordado entre a *Scrum Master* e o Time que uma parte do tempo do próximo *Sprint* seria utilizada para a realização de um treinamento sobre o conteúdo proposto. Porém, antes disso, o Time deveria iniciar os seus estudos e tentar fazer alguns exercícios que seriam enviados para o e-mail de cada membro, podendo contar ainda com o suporte remoto da *Scrum Master* e do membro colaborador externo. A decisão tomada pela *Scrum Master* levou em consideração que a falta de conhecimento deste conteúdo poderia inviabilizar a continuidade do projeto. Assim, seria melhor atrasá-lo do que finalizá-lo naquele momento.

No dia 12/04/17, conforme combinado, o grupo se reuniu para tirar as dúvidas sobre os exercícios propostos deixados na última reunião. Durante a semana, quase todos os membros solicitaram auxílio para a realização das suas atividades. O treinamento/tira-dúvidas foi realizado e, devido aos vários questionamentos levantados pelo grupo, acabou demorando mais do que o previsto, tomando quase que todo o tempo disponível. Os poucos minutos restantes deste dia foram utilizados para a retomada da *Sprint Planning* do *Sprint 2*. Agora, com as dúvidas sanadas, o Time se mostrou mais confiante para executar as atividades propostas. Após a apresentação das tarefas, foi dada ao Time a autonomia para escolherem livremente aquelas que gostariam de trabalhar. A codificação foi feita remotamente e acompanhada pelo Trello.

A continuidade do *Sprint 2* foi realizada no dia 19/04/2017. O Time iniciou atualizando o quadro *Kanban* conforme o Trello. Durante a *Sprint Review* as funcionalidades de cada membro foram apresentadas e testadas pela *Scrum Master* e discutidas entre o Time. Foram detectados pequenos erros na realização de algumas tarefas, após a integração de todos os códigos, evidenciando a ausência de testes mais eficazes pós-integração. Houve também o entendimento ambíguo do que deveria ser feito em algumas tarefas, alertando para um cuidado maior no processo de comunicação entre o Time. As falhas encontradas foram registradas pelo Time para que fossem insumos para o próximo *Sprint* e assim, corrigidas. O trabalho do Time para esta história pode ser conferido na Figura 16.

Figura 16 - Tela de pesquisa de exercícios (História 2- Etapa 1)

Pesquise exercícios por disciplina, termo ou ambos:

Escolha o tipo de pesquisa:

Disciplina Termo Disciplina e termo

Disciplina:

Termo:

Fonte: Própria autora.

Na *Retrospective Review* o Time expôs uma facilidade maior no entendimento do código quando integrado, com o uso da padronização, porém, reconheceram que avanços ainda deveriam ocorrer neste sentido. Outro ponto positivo apontado foi a liberdade dada aos membros na escolha das tarefas que deveriam implementar, trazendo mais conforto ao trabalho realizado. Mais uma vez, boa parte do Time não se atentou para o registro do tempo gasto no desenvolvimento das suas atividades, comprometendo a previsibilidade de trabalhos futuros. Nesta reunião, algumas melhorias ainda foram sugeridas, como:

- Criar um grupo no *Whatsapp* para auxiliar na comunicação remota entre os membros, já que através do Trello isso não era possível.
- Definir um modo de teste antecipado às entregas realizadas para a *Scrum Master/Product Owner*.
- Melhorar a comunicação entre o Time para evitar ambiguidade no entendimento do que deveria ser feito em cada tarefa.
- Por fim, realizar a rotatividade do membro responsável por integrar o código para que todos pudessem passar por este processo.

Devido ao atraso ocorrido no projeto por dificuldades na execução da história 2, para não prejudicar o planejamento realizado, a próxima história (permitir a criação de listas de

exercícios) que seria considerada no *Sprint 3*, precisou ser suprimida. Por causa disso, uma nova história foi alocada para o *Sprint 3* (definir o *layout* e realizar a integração do *software*). Apesar de que o cancelamento da história mencionada não fosse algo desejável, a *Scrum Master* entendeu que esta seria a melhor decisão a ser tomada naquele momento. Assim, o objetivo da Etapa1 que era de realizar a capacitação do Time, não seria comprometido.

Na sequência deste dia (19/04/17), a *Sprint Planning* do *Sprint 3* foi realizada. A história desse *Sprint* (Quadro 15) foi apresentada ao Time e as tarefas foram divididas e discutidas entre os membros. Juntos, o Time modelou as *interfaces* gráficas que seriam construídas para o *software*.

Quadro 15 - História de Usuário do *Sprint 3*

ID:	3	Prioridade:	Alta	Sprint:	3
Nome da História:	Criar <i>interface</i> gráfica do sistema				
Descrição:	Como usuário desejo utilizar as funções do sistema através de uma interface gráfica, contendo: tela de cadastro de exercícios, tela de pesquisa dos exercícios e tela que mostre os resultados das pesquisas realizadas para que eu possa efetivamente utilizar o <i>software</i> .				
Pontos estimados:		Pontos consumidos:			

Fonte: Própria autora.

As reuniões de Revisão e Retrospectiva do *Sprint 3* foram realizadas no dia 26/04/17. Como previsto, o quadro *Kanban* foi atualizado conforme o Trello. A *Sprint Review* mostrou que, devido às correções identificadas no *Sprint 2*, que se acumularam para o *Sprint 3*, o tempo para o desenvolvimento das tarefas deste *Sprint* foi insuficiente. Ao receber e testar as tarefas entregues pelo Time, a *Scrum Master* identificou ainda alguns erros decorrentes da falta de padronização, porém, numa frequência cada vez menor. Uma das tarefas pendentes continuou sem solução, a *Scrum Master* orientou que esta pendência fosse resolvida.

Na Retrospectiva a equipe detectou como principal melhoria a ser realizada um ajuste no planejamento todas as vezes em que houvessem pendências do *Sprint* anterior, evitando acúmulo de tarefas e, conseqüentemente, atrasos no projeto. O tempo restante deste dia foi dedicado para o trabalho com as tarefas que não haviam sido atendidas.

No último dia do projeto (28/04/17) o Time apresentou a versão final do trabalho desenvolvido. Durante os testes feitos pela *Scrum Master* alguns erros ainda foram detectados

devido à imaturidade da equipe com as ferramentas de integração do *software*. Apesar de apresentar algumas falhas, todas as funcionalidades propostas para o *software* foram implementadas pelo Time. Após este momento os alunos foram convidados a fornecerem um *feedback* sobre o projeto realizado.

O Quadro 16 apresenta a sequência cronológica de como as atividades foram efetivamente realizadas durante a Etapa 1 deste projeto.

Quadro 16 - Planejamento realizado (Etapa 1)

Data	Sprint	Eventos	Requisitos
15/03/17		<i>Sprint Planning</i> do Projeto (2h)	
22/03/17		Cancelado	
29/03/17	1	<i>Sprint Planning</i> do <i>Sprint1</i> (1h). Execução do <i>Sprint1</i> (1h_Desenvolvimento/ Suporte presencial). (Mais 4 horas remotamente)	Permitir o cadastro de exercícios didáticos.
05/04/17	1 e 2	Reuniões de Revisão e Retrospectiva do <i>Sprint1</i> (1h). <i>Sprint Planning</i> do <i>Sprint 2</i> (1h). (Mais 4 horas remotamente)	Permitir a consulta dos exercícios didáticos cadastrados.
12/04/17	2	Treinamento Banco de Dados. (1h30min) <i>Sprint Planning</i> do <i>Sprint 2</i> (30min).	Definir o <i>layout</i> e realizar a integração do <i>software</i> .
19/04/17	3	Reuniões de Revisão e Retrospectiva da <i>Sprint 2</i> (1h). <i>Sprint Planning</i> da <i>Sprint 3</i> (1h). (Mais 4 horas remotamente)	
26/04/17	3	Reuniões de Revisão e Retrospectiva da <i>Sprint 3</i> (1h). Suporte para integração e apresentação final do <i>software</i> .	
28/04/17		Apresentação do <i>software</i> .	

Fonte: Própria autora.

5.2.3 Meta-fase: Etapa 1

Analisando a descrição do ciclo de condução da Etapa 1 é possível perceber que várias adaptações foram necessárias para que o *framework Scrum* se adequasse à realidade da equipe do Nosso Exercício. O Quadro 17 demonstra um comparativo entre a estrutura original do *framework*, conforme é sugerida comumente na bibliografia e como foi efetivamente utilizado nesta etapa.

Quadro 17 - Adaptações do Scrum feitas na realidade do Nosso Exercício

Recomendações do Scrum...	Como foi realizado...
Equipes multidisciplinares com habilidades e conhecimentos para desenvolver o <i>software</i> de ponta a ponta. (SABBAGH, 2016).	O conhecimento da equipe era heterogêneo, alguns membros possuíam pouca experiência em desenvolvimento de <i>software</i> .
Equipes trabalham no mesmo ambiente físico, 8 horas por dia.	O Time trabalhou presencialmente e remotamente, 6 horas por semana.
Tamanho ideal para equipes <i>Scrum</i> : 6 a 9 componentes.	Tamanho da equipe formada: 7 componentes.
<i>Sprints</i> de tamanho fixo (de 1 a 4 semanas).	<i>Sprints</i> de tamanho fixo (de 1 semana).
O <i>Product Backlog</i> é criado e priorizado pelo <i>Product Owner</i> .	Apesar de ter sido modificado para se adaptar ao nível de conhecimento do Time, o <i>Product Backlog</i> foi criado e priorizado pelo <i>Product Owner/Scrum Master</i> .
Três papéis são definidos: <i>Product Owner</i> , <i>Scrum Master</i> e Time de Desenvolvimento.	Os três papéis foram definidos, porém, a <i>Scrum Master</i> teve que assumir também o papel do <i>Product Owner</i> .
Reuniões de planejamento do <i>Sprint</i> devem acontecer no início de cada <i>Sprint</i> .	Conforme original.
Reuniões Diárias acontecem no início de cada jornada de trabalho.	Optou-se por não realizar as reuniões diárias já que a maior parte do desenvolvimento foi realizado remotamente.
Reuniões de Revisão e Retrospectiva acontecem no final de cada <i>Sprint</i> .	Conforme original.
O Time deve estimar o tempo de desenvolvimento para cada história/tarefa.	Não foi cumprido. Alguns membros não registravam o tempo gasto nas tarefas desenvolvidas, além disso, o tempo disponível nas reuniões presenciais quase nunca eram suficientes para a equipe levantar esta discussão.
Os itens do <i>Product Backlog</i> são transformados em histórias de usuários.	Atendido. As histórias eram apresentadas para o Time e divididas em tarefas entre os membros.

Fonte: Própria autora.

O planejamento traçado não foi fielmente cumprido. Ao todo, foram planejados 4 *Sprints*, porém, apenas 3 foram executados. Além disso, uma história precisou ser cancelada.

Inicialmente, alguns encontros presenciais haviam sido programados para que o Time tivesse uma hora dedicada às reuniões e uma hora para o desenvolvimento do *software*. A ideia era promover aos membros a experiência de um trabalho em equipe presencial, colaborativo e de fácil comunicação. Já no *Sprint 1* percebeu-se que essa estratégia não poderia

ser adotada, pois, mesmo sendo realizada a *Sprint Planning* onde as tarefas foram divididas e discutidas entre o Time, o tempo que seria destinado à prática foi tomado por dúvidas em relação às tecnologias utilizadas. Ademais, alguns membros também se perderam na execução dos seus papéis, conforme prevê o *Scrum*. Assim, a meta almejada para este *Sprint* não foi alcançada.

A partir disso, os encontros foram basicamente utilizados para as reuniões. Durante as reuniões de Planejamento, as histórias eram apresentadas ao Time, desmembradas em tarefas, discutidas e distribuídas entre os membros. Neste momento também, a *Scrum Master* e o colaborador externo repassavam as orientações técnicas de como desenvolvê-las.

Nas reuniões de Revisão e Retrospectiva os pontos fortes e fracos do projeto eram desvendados, contribuindo para um aprendizado coletivo contínuo. Durante as reuniões de Revisão, cada membro apresentava para a *Scrum Master* o seu trabalho realizado remotamente, explicando-o de forma objetiva para todo o grupo, compartilhando o conhecimento com todos. Nas reuniões de Retrospectiva dos *Sprints* os membros reviam as práticas que contribuíram positivamente para a produtividade do grupo, que deveriam ser mantidas e, ainda, aquelas que interferiram negativamente devendo ser eliminadas ou corrigidas.

Os pontos fracos mais citados pelo Time durante as reuniões de revisão foram: ausência ou baixa qualidade dos testes realizados no *software*; necessidade de melhoria da padronização do código, com intuito de deixá-lo mais legível e de fácil compreensão para todos os membros; dificuldades na comunicação entre o grupo, ocasionando entendimentos ambíguos sobre as tarefas a serem desenvolvidas; dificuldades com a tecnologia adotada.

Como a cada *Sprint* estas mesmas deficiências eram relatadas, a equipe foi, aos poucos, criando estratégias para contorná-las. Assim, a comunicação remota tornou-se mais eficiente através do uso do Trello, *e-mail* e *Whatsapp*, mantendo o grupo em sintonia constante. Ficou estabelecido que, ao final de cada *Sprint*, antes de apresentar o trabalho realizado para a *Scrum Master*, de forma alternada, um membro do Time iria integrar o *software* e testá-lo, a fim de melhorar a qualidade do produto final entregue. O próprio Time foi aprimorando seu mecanismo de padronização do código, cobrando uns dos outros a notação definida por eles. A cada *Sprint*, as dificuldades com a tecnologia eram amenizadas através da prática e melhoria na comunicação realizada.

Como proposta inicial, pode-se concluir, a partir do *feedback* dos alunos, que a implantação do *Scrum* na Etapa 1 deste trabalho cumpriu com o seu objetivo que era de capacitar a equipe em relação ao uso das tecnologias *web* já mencionadas e do *framework* proposto. Ideal seria se este mesmo grupo tivesse a oportunidade de novamente utilizar o

framework, agora, com uma visão mais madura sobre o mesmo, para que os objetivos de evolução no desenvolvimento do Nosso Exercício fossem alcançados.

Resultados semelhantes podem ser encontrados no estudo realizado por Rocha *et al.* (2017). Neste trabalho o *Scrum* também foi utilizado por alunos em formação acadêmica como ferramenta de gestão de projetos de *software*. Buscou-se analisar se o *framework* contribuiria para uma maior agilidade no desenvolvimento de *software* e na integração entre a equipe. A análise deste estudo comprovou que, além de contribuir para a gestão dos projetos, melhorando a comunicação e a evolução dos *Sprints*, o *Scrum* promoveu uma maior autonomia dos estudantes em relação ao aprendizado e tomadas de decisão.

5.3 Etapa 2: Uso do *Scrum* no desenvolvimento do Nosso Exercício

5.3.1 Fase Preliminar da Etapa 2

Conforme já mencionado, a capacitação realizada na Etapa 1 consistiu numa tentativa de preparar os alunos do PET para o desenvolvimento das novas funcionalidades para o *website* Nosso Exercício, registradas no *Product Backlog*. Apesar de terem relatado ganhos de aprendizado nas tecnologias de desenvolvimento, bem como no uso do *Scrum*, percebeu-se que estes alunos ainda não estavam maduros o suficiente para darem continuidade ao projeto. Isso porque, como este se transformou no trabalho de mestrado desta pesquisadora, o tempo tornou-se um fator relevante para que seus objetivos fossem cumpridos.

Diante disso, uma adaptação ao Time *Scrum* fez-se necessária para que o desenvolvimento do Nosso Exercício pudesse efetivamente ser realizado. O novo Time *Scrum* foi composto por apenas três integrantes, a saber: esta pesquisadora no papel de *Scrum Master*, o membro colaborador externo como desenvolvedor e o professor coordenador como *Product Owner*. Após conversas com os responsáveis pelo projeto, houve o entendimento de que este Time estaria mais preparado para a realização do trabalho proposto dentro do tempo disponível.

Conforme o Guia do *Scrum* 2016, menos de três integrantes no Time de Desenvolvimento podem diminuir a interação e resultar em um menor ganho de produtividade no projeto (SCHWABER; SUTHERLAND, 2016), porém, estudos como os de Pagotto (2016) mostram que é possível obter benefícios do *Scrum* para o desenvolvedor solo (individual). O autor orienta neste caso, que as reuniões diárias não são passíveis de acontecer, todavia, os eventos e artefatos do *Scrum* original podem ser mantidos com suas devidas adaptações.

A Etapa 2 deste projeto objetivou efetivamente o desenvolvimento das novas funcionalidades desejáveis para o Nosso Exercício. O ciclo de condução desta etapa será descrito na próxima seção.

5.3.2 Ciclo de Condução: Etapa 2

A *Sprint Planning* desta etapa aconteceu no dia 17/05/17 e nela algumas decisões importantes foram tomadas. Ficou acordado que o trabalho do desenvolvedor seria realizado remotamente e as reuniões parcialmente presenciais, pois, o *Product Owner* ainda encontrava-se residindo em outra cidade. Nos dias das reuniões, a *Scrum Master* e o desenvolvedor se reuniam no mesmo local e através de videoconferência, comunicavam-se com o *Product Owner*.

Para viabilizar a interação do Time, foi mantido o uso do Trello e extinto o quadro *Kanban* físico, uma vez que os membros se concentrariam no projeto muito mais remotamente do que presencialmente. Além disso a comunicação era realizada através do *Skype*¹³, *e-mail*, *Whatsapp* e *Bitbucket*. Este último foi utilizado para compartilhar o código implementado, criando um repositório para o projeto.

O conceito de ‘Pronto’ foi definido, considerando que para uma funcionalidade estar ‘pronta’ ela deveria: ser codificada, passar nos critérios e testes de aceitação, passar nos testes feitos pelo *Product Owner*, conforme sugere [Sabbagh](#) (2016). Basicamente este conceito se diferenciou da Etapa 1 pela presença dos critérios e testes de aceitação e pelo fato de que agora o *Product Owner* estaria presente para aceitar ou não as partes do *software* desenvolvido. Além disso, como apenas um desenvolvedor era responsável pelo código, não havia integração de tarefas.

As reuniões diárias também não foram realizadas, já que a equipe não estaria reunida presencialmente durante as atividades de desenvolvimento do *software*.

A princípio, os *Sprints* da Etapa 2 foram pensados com o tempo fixo de uma semana. Porém, já na execução do *Sprint 1*, o Time percebeu que o tempo definido seria insuficiente, pois, baseado na experiência anterior, sabia-se que a falta de domínio com a tecnologia, neste caso o Drupal, poderia interferir no tempo de implementação das histórias. Por causa disso, foram definidos *Sprints* de 15 dias, contemplando 8 horas de desenvolvimento

¹³ *Software* que permite que duas ou mais pessoas no mundo todo conversem entre si, usando a internet, através de chamadas de vídeo e voz gratuitas. Também é possível enviar mensagens de *chats* e compartilhar arquivos, usando celular, computador ou *tablet*. (MICROSOFT, 2017).

e 3 horas destinadas para as reuniões. A definição deste tempo também levou em consideração a disponibilidade do Time, que se ocupava com outros afazeres paralelos ao projeto.

O planejamento inicial da Etapa 2, com as datas para os eventos do *Scrum* e as histórias trabalhadas em cada *Sprint* (*Sprint Backlog*) pode ser conferido no Quadro 18. Foram planejados seis *Sprints* e, em cada um, duas histórias seriam contempladas. Importante mencionar que a divisão das histórias de usuários em cada *Sprint* não foi determinada por uma priorização feita pelo *Product Owner* conforme é sugerido no *Scrum* original. Por considerar todas as histórias com prioridade alta e não estabelecer uma ordem específica para o desenvolvimento, o *Product Owner* deu autonomia para a *Scrum Master* e o desenvolvedor tomarem esta decisão. Assim, o critério utilizado baseou-se no nível de complexidade e de dependência entre as histórias.

Quadro 18 - Planejamento Inicial da Etapa 2 do projeto

<i>Sprint N°</i>	<i>Sprint Planning / Refinamento do Product Backlog</i>	Histórias (ID)	Reuniões de Revisão e Retrospectiva
1	24/05	7 e 9	07/06
2	07/06	6 e 1	21/06
3	21/06	4 e 10	05/07
4	05/07	5 e 11	19/07
5	19/07	2 e 8	02/08
6	02/08	3	16/08

Fonte: Própria autora.

O *Product Backlog* foi revisto e os requisitos levantados foram transformados em histórias de usuários à medida em que estas eram consideradas em cada *Sprint*. Como o *Product Backlog* é dinâmico, durante o desenvolvimento deste trabalho, algumas modificações foram realizadas no mesmo. O Quadro 19 demonstra uma visão completa do *Product Backlog* ao final da etapa.

Quadro 19 - *Product Backlog* utilizado na Etapa 2 do projeto

<i>Product Backlog - Projeto Nosso Exercício</i>		
ID	História	Classificação
1	Como usuário autenticado, quero ter a opção de realizar <i>download</i> dos exercícios e/ou listas de exercícios nos formatos .pdf (<i>Adobe</i>), .odt (<i>Writer-Libre Office</i>), .tex (<i>Latex</i>), e .doc (<i>Word</i>) para que os exercícios já venham formatados como estão no <i>site</i> .	Nova Funcionalidade
2	Como usuário autenticado, ao criar uma lista, quero escolher a quantidade de exercícios que devem compor esta lista.	Nova Funcionalidade
3	Como usuário autenticado, quero ter a opção de resolver questões de múltipla escolha no sistema e obter o resultado para que eu possa treinar os meus estudos.	Nova Funcionalidade
4	CANCELADA	-
5	CANCELADA	-
6	Como usuário autenticado, navegando em uma lista pública, desejo ter a opção visualizar completamente as informações de cada exercício da lista.	Melhoria
7	Como usuário autenticado ou registrado, desejo visualizar na tela inicial, sugestões de disciplinas já cadastradas no <i>site</i> e realizar a busca por exercícios, clicando em uma destas opções, para que eu possa ter mais de uma forma de realizar uma busca no sistema.	Melhoria
8	Como <i>Product Owner</i> desejo obter melhorias no <i>layout</i> /usabilidade do <i>site</i> para proporcionar ao usuário uma melhor experiência de navegação.	Melhoria
9	Como usuário anônimo ou autenticado, ao realizar uma busca e inserir um termo e/ou disciplina, desejo que o <i>software</i> retorne apenas os exercícios ‘com resolução’ e/ou ‘comentários’, para que eu possa criar uma lista ou consultar os exercícios que atendam a estes critérios.	Melhoria
10	Como usuário anônimo ou autenticado, ao realizar uma pesquisa, quero que o sistema retorne a quantidade de itens resultantes da busca.	Melhoria
11	CANCELADA	-
12	Como administrador, quando o usuário registrado clicar nos <i>links</i> “Denunciar erro” e/ou “Denunciar infração”, quero que o sistema envie uma mensagem para que o usuário confirme a sua ação, gerando maior confiabilidade à denúncia.	Melhoria

Fonte: Própria autora.

Por decisão do *Product Owner*, após diversas análises e discussões com a *Scrum Master* e o desenvolvedor, as histórias 4, 5 e 11 foram canceladas. Em contrapartida, a história 12 foi acrescentada como uma funcionalidade a ser melhorada.

A *Sprint Planning* do *Sprint1* da Etapa 2 foi realizada no dia 24/05/17 e considerou os itens 7 e 9 do *Product Backlog*. Nesta reunião a *Scrum Master* e o desenvolvedor discutiram detalhadamente com o *Product Owner* as funcionalidades que deveriam ser implementadas, com intuito de buscarem uma compreensão maior do trabalho a ser realizado. Após esta conversa, as histórias de usuário para este *Sprint* foram criadas. Como o *Product Owner* era o membro com maior conhecimento sobre o Nosso Exercício e também sobre o Drupal, este teve um papel extremamente relevante nesta etapa do projeto.

A história de usuário criada para o item 7 do *Product Backlog* pode ser visualizada no Quadro 20.

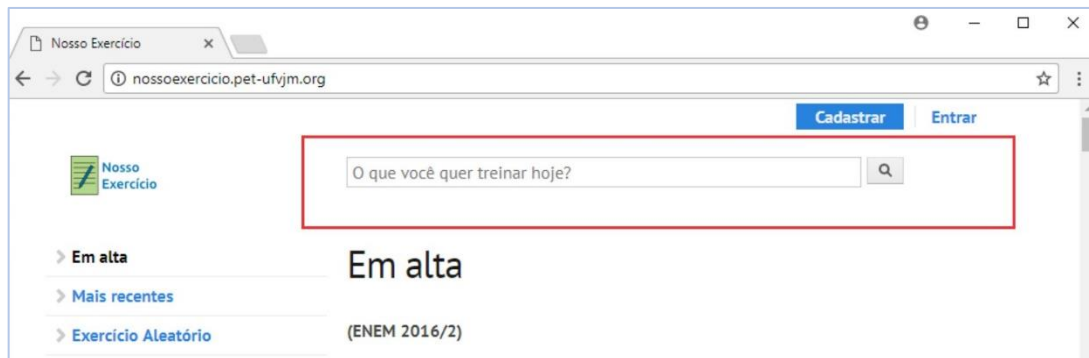
Quadro 20 - Descrição da história de usuário 7- Sprint 1 (Etapa 2)

ID da História: 7	Prioridade: Alta	Sprint: 1
Nome da História:	Mostrar opções de disciplinas existentes ao realizar uma busca	
Descrição:		
<p>Como usuário anônimo ou autenticado, desejo visualizar na tela inicial, sugestões de disciplinas já cadastradas no <i>site</i> para pesquisar por exercícios, clicando em uma destas opções, para que eu possa ter mais de uma forma de realizar uma busca no sistema.</p>		
Pontos estimados:		Pontos consumidos:

Fonte: Própria autora.

À fim de esclarecer a necessidade desta modificação no sistema, é mostrada na Figura 17 a tela inicial do Nosso Exercício na sua versão original, disponibilizando apenas uma opção para o usuário que desejasse realizar uma busca (destaque em vermelho). Entendendo que desta forma o sistema contribuía apenas com aqueles que já sabiam o termo pelo qual deveriam pesquisar, o *Product Owner* solicitou uma nova opção de pesquisa, mais intuitiva, que sugerisse para o usuário as opções de disciplinas já disponíveis.

Figura 17 - Busca por exercícios antes da implementação da história 7



Fonte: Própria autora.

Por sugestão da *Scrum Master*, após a obtenção de todos os detalhes para o desenvolvimento das funcionalidades do sistema, foram criados critérios e testes de aceitação para cada história, com o intuito de orientar e validar o trabalho do desenvolvedor e gerar maior valor para o cliente. Os critérios e testes de aceitação criados para a história 7 podem ser conferidos nos Quadros 21 e 22.

Quadro 21 - Critério de Aceitação da história de usuário 7

Critério(s) de Aceitação	
História:	7
Critério 1: Na página inicial as pesquisas por exercícios deverão ser realizadas digitando o termo diretamente no campo de busca, ou, selecionando um dos <i>links</i> com os nomes das disciplinas que ficarão disponíveis abaixo do campo de busca. Ao digitar um termo no campo de busca ou clicar em dos <i>links</i> com os nomes das disciplinas, o sistema deverá retornar os exercícios correspondentes à busca.	

Fonte: Própria autora.

A descrição deste critério de aceitação auxiliou o desenvolvedor a compreender a funcionalidade com a mesma visão do *Product Owner*. Já o teste de aceitação foi utilizado para que o desenvolvedor, ao implementar a história, conhecesse qual comportamento o sistema deveria assumir e, ao final da implementação, executar o teste com intuito de validar ou não o resultado do seu trabalho. Deste modo foi possível eliminar vários erros que poderiam surgir no sistema.

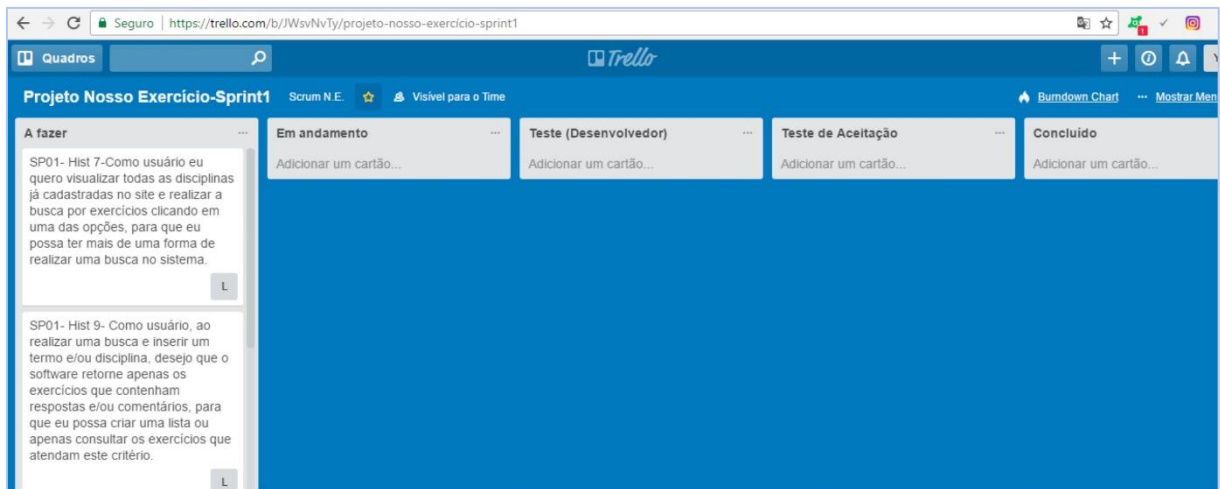
Quadro 22 - Teste de Aceitação da história de usuário 7

Teste de Aceitação		
História:	7	
Número	Ação	Resultado Esperado
1	- O usuário digita um termo no campo de busca.	- O sistema retorna todos os exercícios que se relacionam com o termo digitado.
2	- O usuário clica no <i>link</i> do nome de uma disciplina.	- O sistema retorna todos os exercícios que se relacionam com o <i>link</i> selecionado.

Fonte: Própria autora.

A Figura 18 mostra o quadro criado no Trello para o Projeto Nosso Exercício nesta etapa, utilizado para promover transparência ao Time do andamento das histórias. Demonstra uma visão inicial do *Sprint1* com as histórias que foram desenvolvidas neste ciclo (histórias 9 e 7) , cada uma registrada em um cartão na coluna correspondente ao seu *status*.

Figura 18 - Quadro no Trello indicando o início do *Sprint 1*

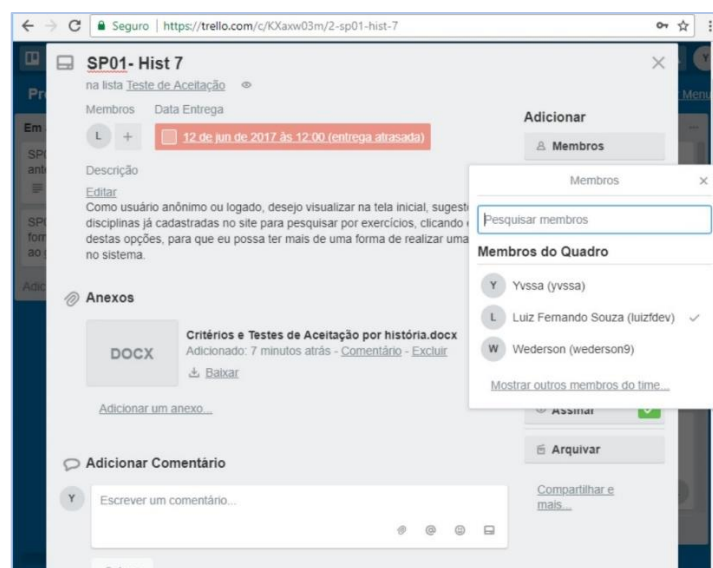


Fonte: Própria autora.

O quadro foi dividido em 5 colunas: a fazer (histórias de um ou mais *Sprints* a serem desenvolvidas); Em andamento (histórias em desenvolvimento em um determinado *Sprint*). Teste Desenvolvedor (histórias que estão sendo testadas pelo desenvolvedor e/ou *Scrum Master*); Teste de Aceitação (histórias aptas para o *Product Owner* testar); Concluído (histórias validadas pelo *Product Owner*). À medida que o desenvolvedor realizava o seu trabalho, os cartões no Trello iam avançando de uma coluna para outra.

Cada cartão armazenava detalhes da história que o representava, como o seu número identificador e o *Sprint* ao qual pertencia, a sua descrição, se a entrega estava dentro ou fora do prazo, os membros autorizados a visualizá-lo, dentre outras informações. Além disso, era possível anexar a cada cartão os critérios e testes de aceitação de cada história (Figura 19).

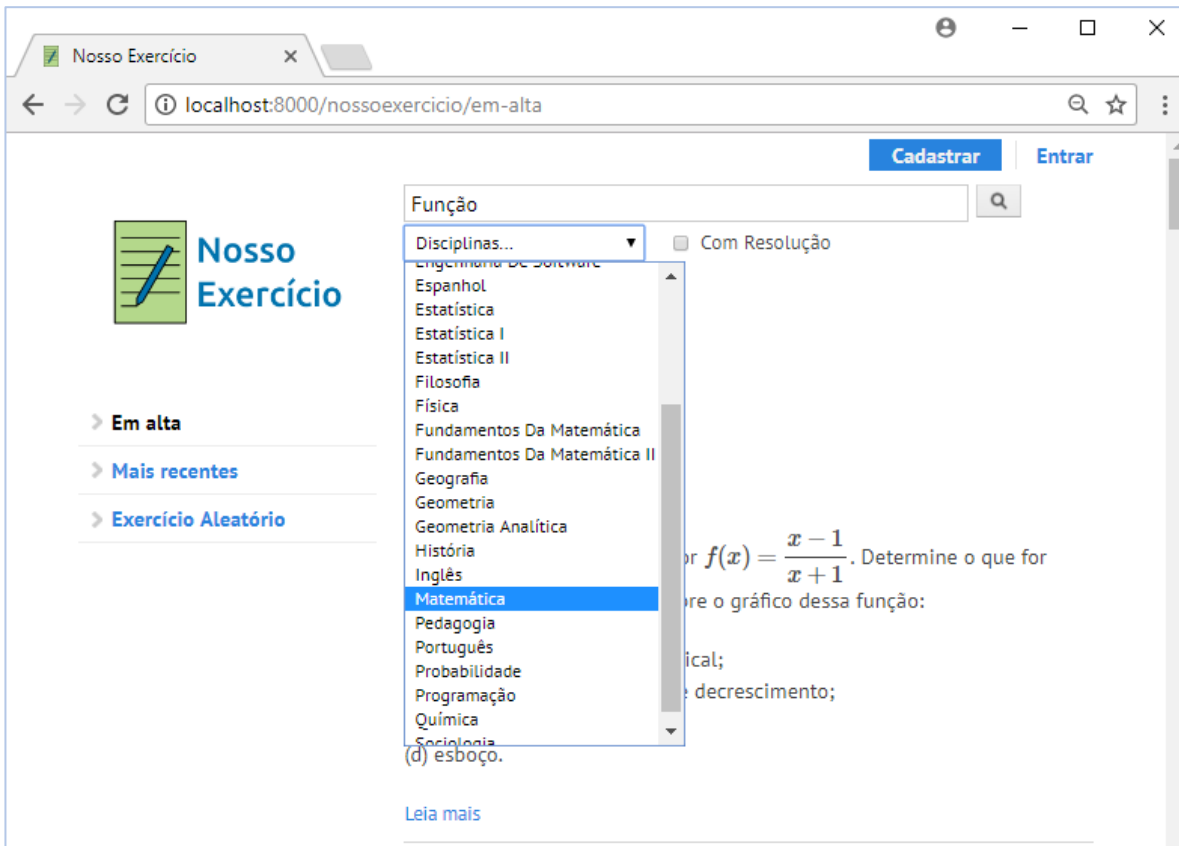
Figura 19- Detalhes do cartão da história 7 no Trello



Fonte: Própria autora.

A Figura 20 apresenta a tela do Nosso Exercício após o desenvolvimento da história 7. Agora, a busca poderá ser realizada digitando um termo na barra de pesquisa e/ou pesquisando por uma disciplina existente.

Figura 20 - Busca por exercícios após implementação da história 7



Fonte: Própria autora.

A história 9 também foi considerada no *Sprint 1*, a sua descrição é demonstrada no Quadro 23.

Quadro 23 - Descrição da história de usuário 9 - *Sprint 1* (Etapa 2)

ID da História:	9	Prioridade:	Alta	Sprint:	1
Nome da História:	Realizar buscas apenas por exercícios que contenham respostas e/ou comentários.				
Descrição:	<p>Como usuário anônimo ou autenticado, ao realizar uma busca e inserir um termo e/ou disciplina, desejo que o <i>software</i> retorne apenas os exercícios ‘com resolução’ e/ou ‘comentários’, para que eu possa criar uma lista ou consultar os exercícios que atendam a estes critérios.</p>				
Pontos estimados:		Pontos consumidos:	2h		

Fonte: Própria autora.

Esta história visou atender uma solicitação do *Product Owner* que era de oferecer ao usuário do sistema uma opção de filtro na realização de buscas por exercícios. Antes da implementação da mesma, a busca retornava todos os exercícios cadastrados, com ou sem resolução e/ou comentários, não permitindo ao usuário interferir neste processo. O Quadro 24 mostra o critério de aceitação para esta história.

Quadro 24 - Critério de Aceitação da história de usuário 9 - *Sprint 1* (Etapa 2)

Critério(s) de Aceitação	
História:	9
Critério 1: Na página inicial, ao realizar uma busca, ao habilitar o campo 'com resolução' e/ou 'com comentários' e confirmar esta busca, o sistema deverá mostrar apenas exercícios que contenham resposta e/ou comentários.	

Fonte: Própria autora.

O teste de aceitação criado para a história 9 pode ser conferido no Quadro 25.

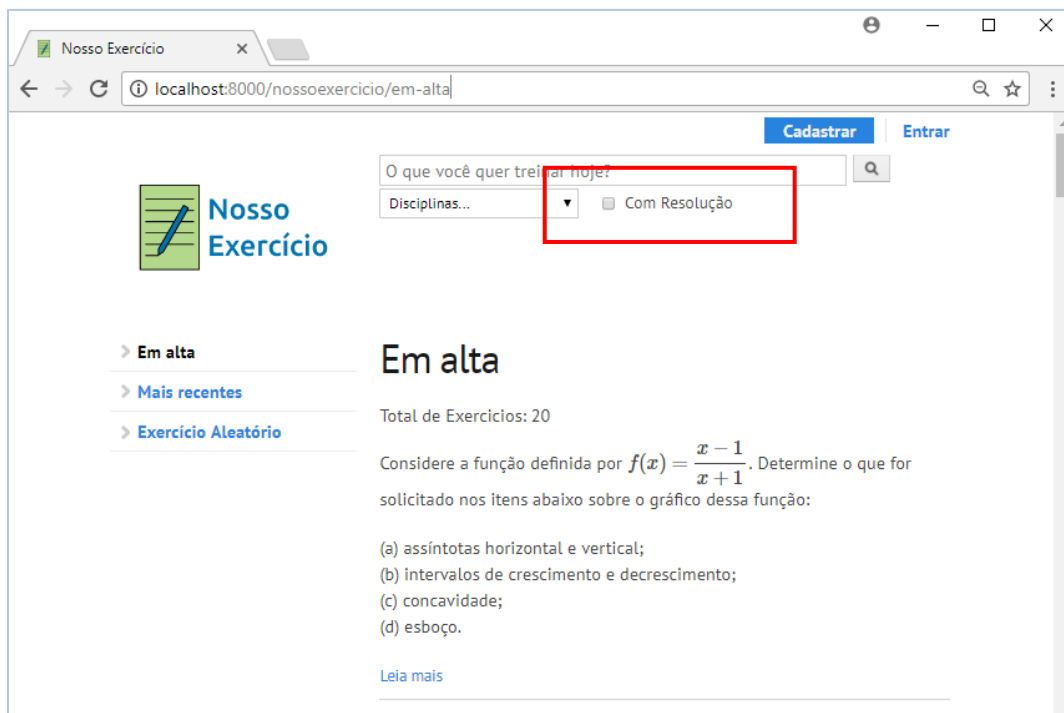
Quadro 25 - Teste de Aceitação da história e usuário 9- *Sprint 1* (Etapa 2)

Teste de Aceitação		
História:	9	
Número	Ação	Resultado Esperado
1	<ul style="list-style-type: none"> - O usuário digita um termo no campo de busca e seleciona a opção 'Com resolução' e/ou 'Com comentários'. - O usuário clica no ícone da lupa ou em um <i>link</i> de uma disciplina para confirmar a busca. 	- O sistema retorna todos os exercícios que se relacionam com o termo digitado, que contenham resolução e/ou comentários .

Fonte: Própria autora.

A Figura 21 demonstra a tela inicial do Nosso Exercício após o desenvolvimento da história 9. Além de usar a barra de pesquisa e/ou selecionar uma disciplina existente, o usuário poderá filtrar ainda mais a sua pesquisa clicando na opção 'Com resolução' (destaque em vermelho).

Figura 21 - Tela do Nosso Exercício com a opção de buscar exercícios apenas ‘Com Resolução’



Fonte: Própria autora.

As Reuniões de Revisão e Retrospectiva do *Sprint1* foram realizadas no dia 31/05/17. Durante a *Sprint Review*, a *Scrum Master* e o desenvolvedor apresentaram as funcionalidades desenvolvidas para serem testadas pelo *Product Owner* que forneceu o seguinte *feedback* das mesmas:

História 7 - o *Product Owner* aprovou parcialmente esta história, pois, confirmou que o seu funcionamento estava correto, porém, sugeriu uma alteração no *layout* criado para a mesma. A *Scrum Master* lembrou ao *Product Owner* que mudanças no *layout* e usabilidade estavam previstas para a história 8. Como melhoria futura para esta funcionalidade foi sugerida uma opção de busca pelas 10 disciplinas mais acessadas no *website*.

A análise mais aprofundada desta história evidenciou a inserção de termos redundantes e erros ortográficos nos nomes das disciplinas já cadastradas no banco de dados do sistema. Percebeu-se que, corrigi-la manualmente seria inviável e que levaria um tempo além do disponível neste *Sprint*, sendo assim esta funcionalidade foi desmembrada e inserida no *Product Backlog* como uma nova história para ser tratada posteriormente.

História 9 – Após o teste desta funcionalidade o *Product Owner* chegou à conclusão de que a busca por exercícios ‘com comentários’ não era relevante e resolveu manter apenas a opção de busca por exercícios ‘com resolução’. Esta correção gerou uma tarefa a ser validada no *Sprint2*.

A *Retrospective Review* do *Sprint1* foi realizada logo após a *Sprint Review*. Neste momento, *Scrum Master* e Desenvolvedor discutiram os principais acontecimentos e fatos deste *Sprint*. Como pontos positivos que deveriam ser mantidos, foram identificados o comprometimento do Time e o registro dos critérios e testes de aceitação feitos pela *Scrum Master* para nortear o trabalho do desenvolvedor. Já os pontos negativos relatados, que deveriam ser melhorados, e, as ações esperadas para os próximos *Sprints* foram:

- Ausência de um *software* de controle de versão. A instalação deste *software* visou, além do controle das versões, o compartilhamento do projeto para todos os membros. Ficou definido o *BitBucket* para esta tarefa.

- Inserção de todos os membros do Time no Trello, pois devido a esta falha o mesmo ainda não estava sendo utilizado corretamente.

- Dificuldade de registrar o tempo gasto com o desenvolvimento da funcionalidade nos cartões de história, devido a falta de domínio no uso do Drupal.

- Conforme relato do desenvolvedor, o tempo definido para o *Sprint* foi insuficiente. A dedução feita pelo mesmo considerou que, após o *feedback* do cliente, quase sempre haveria alguma mudança a ser considerada para o próximo *Sprint*, acumulando com as histórias que já deveriam ser implementadas no *Sprint* atual. Além disso, a imaturidade com o Drupal dificultava a compreensão real de como uma funcionalidade deveria ser desenvolvida com exatidão. Assim, ficou decidido que o tamanho dos *Sprints* passaria a ser de 15 dias. Sabbagh (2016) menciona que a modificação da duração do *Sprint* pode acontecer dentro do mesmo projeto, geralmente detectada pelo Time após a *Retrospective Review* e que isso deve ocorrer raramente.

Após encerrar as atividades do *Sprint 1*, neste mesmo dia, aconteceu a *Sprint Planning* do *Sprint 2*, abrangendo as histórias 1 e 6. Como de costume, o Time discutiu sobre os detalhes das novas funcionalidades a serem desenvolvidas. Buscou-se no decorrer deste ciclo, executar as ações estabelecidas na *Retrospective Review* do *Sprint1*, como também realizar as atividades de desenvolvimento das novas histórias e correções pendentes levantadas na última *Sprint Review*. No encerramento deste *Sprint* ocorreram as Reuniões de Revisão e Retrospectiva do *Sprint2*, logo em seguida a *Sprint Planning* do *Sprint 3* e assim, o ciclo permaneceu para os demais *Sprints* até que todos fossem concluídos.

Devido ao número considerável de *Sprints* realizados nesta etapa, para que a descrição deste ciclo de condução não ficasse demasiadamente repetitivo e o foco da análise do *Scrum* neste trabalho fosse mantido, optou-se por detalhar nesta seção apenas o primeiro *Sprint*. Para os outros, serão apresentados nos Quadros 26 e 27 um panorama geral dos aspectos mais

relevantes de cada um, já que os ciclos de desenvolvimento utilizando o *Scrum* estabelecem a mesma sequência de atividades para todos. Ademais, nos apêndices de A a D deste trabalho o leitor poderá conferir todas as histórias, as justificativas para as mesmas, os critérios e testes de aceitação criados, bem como as alterações no *layout* do *website* Nosso Exercício depois do desenvolvimento deste projeto.

O Quadro 26 revela como os *Sprints* foram efetivamente realizados na Etapa 2, as divisões das histórias por *Sprint* e as datas das reuniões. Foram realizados 8 *Sprints* e não 6 como planejado, estendendo o prazo de finalização do projeto. Mesmo ajustando o tamanho dos *Sprints* para 15 dias, houveram atrasos em quase todas as entregas.

Quadro 26 - Visão Geral do Ciclo de Condução da Etapa 2

	Histórias	Reunião de Planejamento	Reunião de Revisão e Retrospectiva
<i>Sprint 1</i>	7 e 9	24/05/17	31/05/17
<i>Sprint 2</i>	1 e 6, mais pendências do <i>Sprint 1</i>	31/05/17 Atraso (falta de energia)	21/06/17
<i>Sprint 3</i>	Pendências dos <i>Sprints</i> anteriores	21/06/17 Atraso (dificuldades com o Drupal)	12/07/17 Pausa no projeto
<i>Sprint 4</i>	2, 11 e 12, mais pendências dos <i>Sprints</i> anteriores	08/08/17	22/08/17
<i>Sprint 5</i>	3, 4 e 10 mais pendências dos <i>Sprints</i> anteriores	22/08/17 Atraso (dificuldades com o Drupal)	19/09/17
<i>Sprint 6</i>	3 (continuação) e 5	19/09/17 Atraso (dificuldades com o Drupal)	10/10/17
<i>Sprint 7</i>	8 mais pendências dos <i>Sprints</i> anteriores	10/10/17 Atraso (dificuldades com o Drupal)	31/10/17
<i>Sprint 8</i>	Pendências dos <i>Sprints</i> anteriores	31/10/17	21/11/17

Fonte: Própria autora.

Ao final do último *Sprint* todas as funcionalidades válidas haviam sido codificadas, porém, o projeto não foi finalizado, pois, atividades relacionadas à integração dos incrementos criados e atualização do *software* ainda deveriam ser realizadas. Em conformidade com o *Product Owner*, optou-se por não efetuar este trabalho dentro dos *Sprints*. Sabbagh (2016) considera que esta é uma atitude válida em projetos *Scrum* por entender que para o *Product Owner* nem sempre um incremento representa valor suficiente para ser utilizado por seus usuários, cabendo esta decisão ao mesmo.

5.3.3 Meta-fase: Etapa 2

A análise do ciclo de condução da Etapa 2 deste projeto revelou que várias foram as dificuldades enfrentadas pelo Time *Scrum* no desenvolvimento das novas funcionalidades desejáveis para o *website* Nosso Exercício.

A inexperiência com o Drupal foi considerada o maior gargalo do projeto. Apesar do desenvolvedor já ter realizado vários outros trabalhos utilizando as tecnologias nas quais o Drupal se apoia (PHP, HTML, CSS, MySQL), a falta de domínio com os recursos deste gerenciador de conteúdo provocou atrasos e dificuldades na compreensão de como determinadas funcionalidades seriam desenvolvidas. Cada nova história exigia um tempo de aprendizagem significativo da ferramenta por parte do desenvolvedor, interferindo no prazo definido para sua entrega.

Ainda que o Drupal possua recursos para simplificar a criação de *websites* por usuários leigos, no contexto deste projeto, este CMS demandou um nível de conhecimento mais aprofundado. Para o desenvolvimento de todas as histórias foram utilizadas técnicas de programação *web* trabalhando no *back-end* do sistema. A história de ID7 requereu a criação de um módulo customizado, opção no Drupal geralmente realizada por programadores mais experientes na ferramenta.

A alta curva de aprendizado do Drupal também interferiu no registro das estimativas do projeto. No decorrer dos *Sprints* o desenvolvedor buscou registrar o tempo efetivamente gasto na implementação das funcionalidades, com intuito de preencher o campo ‘pontos consumidos’ dos cartões de história. A ideia era que através deste registro, funcionalidades semelhantes pudessem ser estimadas futuramente por comparação, auxiliando na mensuração do esforço necessário para o trabalho com as mesmas. As frequentes interrupções com pesquisas e busca de soluções sobre a ferramenta, dificultaram este processo.

Durante este ciclo as histórias de usuário foram discutidas nas Reuniões de Planejamento, desenvolvidas (codificadas), entregues para o *Product Owner* fornecer o seu *feedback* nas Reuniões de Revisão e inspecionadas para correções e/ou validações. O Quadro 27 apresenta um apanhado geral do *status* de cada história na condução dos *Sprints*. A legenda a seguir deve ser considerada para a compreensão do mesmo.

H (1 a 12) - Histórias de usuário contempladas na Etapa 2.

SP (1 a 8) - *Sprints* realizados na Etapa 2.

RP - Reunião de Planejamento (*Sprint Planning*).

RR - Reunião de Revisão (*Sprint Review*).

- I- A história foi iniciada em um respectivo *Sprint*.
- AJ- A história foi mostrada ao *Product Owner* e requer ajustes.
- CO - A história foi corrigida conforme os ajustes sugeridos pelo *Product Owner*.
- D- A história é complexa e foi desmembrada em outras histórias.
- V – A história foi validada pelo *Product Owner*.
- V* – A história foi validada pelo *Product Owner*, com pequenas restrições
- CAN – A história foi cancelada pelo *Product Owner*.

Quadro 27 - Visão geral do status das histórias em cada *Sprint* da Etapa 2

	SP1		SP2		SP3		SP4		SP5		SP6		SP7		SP8	
	RP	RR	RP	RR	RP	RR	RP	RR	RP	RR	RP	RR	RP	RR	RP	RR
H1			I	AJ/D	CO	AJ	CO	V*								
H2							I	AJ	CO	V						
H3									I	AJ	CO	AJ	CO	AJ	CO	V
H4									CAN							
H5											CAN					
H6			I	AJ	CO	V*										
H7	I	AJ	CO	AJ/D	CO	V*										
H8													I	AJ	CO	V
H9	I	AJ	CO	V												
H10									I	AJ	CO	V				
H11							CAN									
H12							I	V								

Fonte: Própria autora.

Conforme já dito, a primeira versão do *Product Backlog* (Quadro 7) foi modificada transformando-se no *Product Backlog* da Etapa 2. Nele, os requisitos listados foram convertidos em histórias de usuário à medida em que eram contemplados em cada *Sprint*. Ao todo 9 histórias foram desenvolvidas nesta etapa. Três requisitos foram cancelados pelo *Product Owner* e, por causa disso, não foram convertidos em histórias.

Dos requisitos cancelados, os de ID4 (permitir que o usuário crie grupos, de modo que apenas aquele grupo veja uma certa lista) e ID11 (compartilhar listas com ou sem respostas) foram considerados irrelevantes na visão *Product Owner*, entendendo que não mais gerariam valor para o sistema. Já o de ID5 (gerar rascunho dos exercícios) foi encaminhado novamente para o *Product Backlog* como uma funcionalidade futura a ser considerada em outro momento do projeto, rotulada pelo Time como uma história complexa e extensa para aquele *Sprint*. Estas

decisões eram tomadas sempre que o Time discutia melhor as funcionalidades, refinando o *Product Backlog*.

Outra funcionalidade que voltou para o *Product Backlog* foi a história de ID1 (realizar *download* dos exercícios). A *Sprint Review* desta história provou que o seu desenvolvimento era incompatível com o tamanho de um *Sprint*, assim, o Time desmembrou esta história de forma que naquele momento apenas o *download* em PDF fosse considerado. Isso mostrou uma dificuldade do Time em prever, com exatidão, o esforço necessário para o trabalho com as histórias.

Apenas para a história 12 (enviar mensagem de confirmação para o usuário ao clicar em ‘Denuncia erro’ ou ‘Denunciar infração’) foi possível discutir os seus detalhes na *Sprint Planning* e o *Product Owner* validá-la, sem restrições, na *Sprint Review* dentro do mesmo *Sprint*. Isto significou que esta história foi de baixa complexidade ou que o entendimento sobre a mesma ficou claro para o Time. Em todos os outros casos, foi necessário realizar ajustes ou correções após o *feedback* fornecido, com algumas funcionalidades demandando um tempo maior do que outras. Exemplo disso foram as histórias 1 (realizar *download* dos exercícios), 3 (criar questões de múltipla escolha) e 7 (criar *links* das disciplinas na tela inicial).

Algumas vezes, os ajustes solicitados pelo *Product Owner* após os testes das histórias desenvolvidas eram decorrentes de mudanças de opinião do mesmo, revelando uma incerteza na sua visão sobre uma determinada funcionalidade. Mudanças também eram ocasionadas eventualmente, decorrentes de dificuldades por parte da *Scrum Master* e do desenvolvedor no entendimento do que deveria ser feito. Em outros momentos, estes ajustes derivaram de falhas nos testes feitos. Para todos os casos, estas histórias voltavam para os próximos *Sprints* com as correções a serem realizadas.

A necessidade de constantes alterações/ajustes nas histórias, também influenciou no tempo de entrega do projeto. Buscando evitar um acúmulo de pendências, a *Scrum Master* em conformidade com o *Product Owner*, propôs em quase todos os *Sprints*, reduzir o número de novas histórias a serem desenvolvidas ou, até mesmo, não contemplar nenhuma história nova em alguns casos, para que as pendências fossem resolvidas e não mais se acumulassem.

As histórias 1, 6 e 7 foram aprovadas com restrições denotando que o *Product Owner* aceitou o trabalho do desenvolvedor, porém, pequenos ajustes ainda seriam necessários. Isso acontecia após as histórias já terem percorrido vários *Sprints* passando por ajustes e correções. Por causa disso, o seu entendimento já estava bem claro para o Time, ficando uma pendência mínima para a conclusão das mesmas. Assim, o *Product Owner* considerava aquela

história como validada e o desenvolvedor assumia a responsabilidade de finalizá-la, sem comprometer o andamento do seu trabalho.

Confirmando a teoria de que no *Scrum* o *Product Backlog* nunca está completo, o Quadro 28 mostra uma atualização do mesmo após o andamento deste projeto. Histórias que foram desmembradas, canceladas por falta de clareza ou demandadas a partir de uma nova necessidade que surgia ao longo das discussões entre o Time, foram registradas e atualizadas como novos requisitos no *Product Backlog* com intuito de serem contempladas numa próxima evolução da plataforma.

Quadro 28 - Versão final do *Product Backlog* após o desenvolvimento do projeto

Atualização do <i>Product Backlog</i> - Projeto Nosso Exercício		
ID	Descrição	Classificação
13	Permitir o <i>download</i> das listas de exercícios nos formatos .doc, .latex e .odt.	Melhoria
14	Criar um novo mecanismo de busca que permita ao usuário realizar uma pesquisa das 10 disciplinas mais acessadas no <i>site</i> .	Melhoria
15	Corrigir erros ortográficos e remover a duplicidade dos nomes das disciplinas inseridas no banco de dados.	Melhoria
16	Melhorar a usabilidade/ <i>layout</i> do <i>site</i> .	Melhoria
17	Gravar os exercícios sendo digitados como 'Rascunho', de modo que se algum erro de conexão ocorrer o usuário não perca o seu trabalho.	Nova Funcionalidade
18	Elaborar um manual para os usuários do sistema	Nova Funcionalidade

Fonte: Própria autora.

5.4 Comparação dos principais resultados das Etapas 1 e 2

Após a realização dos dois ciclos de condução envolvendo a implantação do *Scrum* na abordagem do Nosso Exercício, nesta seção, os resultados destas etapas serão confrontados, à fim de que se tenha uma visão geral dos principais aspectos de cada etapa. O Quadro 29 apresenta uma comparação da aplicação do *framework Scrum* nas duas etapas deste projeto.

Quadro 29 - Comparação do uso do *Scrum* nas etapas 1 e 2 deste projeto

	Etapa 1	Etapa 2
Objetivo	Capacitação do Time.	Desenvolvimento das funcionalidades do <i>website</i> .
Duração dos <i>Sprints</i>	1 semana (6 horas de dedicação ao projeto)	2 semanas (11 horas de dedicação ao projeto)
<i>Product Backlog</i>	Adaptado ao nível de conhecimento dos alunos e do contexto do Nosso Exercício.	Criado pelo <i>Product Owner</i> , transformado em histórias de usuários nas <i>Sprints Backlogs</i> , atualizado para novas evoluções do projeto.
Time	Composto por 5 integrantes: <i>Scrum Master/Product Owner</i> , 4 desenvolvedores (1 colaborador externo, 3 alunos do PET).	Composto por 3 integrantes: <i>Scrum Master, Product Owner</i> e um desenvolvedor (colaborador externo)
Encontros	Presenciais e remotos (desenvolvimento remoto em sua maior parte)	Presenciais e remotos (desenvolvimento apenas remoto)
Quadro <i>Kanban</i>	Físico e virtual	Apenas virtual
Conceito de 'Pronto'	Funcionalidade desenvolvida, testada individualmente pelo membro que a fez, testada após a integração com as demais tarefas e, por fim, testada e aprovada pela <i>Scrum Master/Product Owner</i> .	Funcionalidade desenvolvida, passada nos critérios e testes de aceitação, testada e validada pelo <i>Product Owner</i> .
Reuniões Diárias	Não realizadas.	Não realizadas.
Estimativas	Não realizadas.	Não realizadas.
Planejamento	Sofreu atrasos devido à inexperiência do Time com as atividades de desenvolvimento de <i>software</i> .	Sofreu atrasos devido à imaturidade com o ambiente de desenvolvimento utilizado, o <i>Drupal</i> .
Comunicação	Trello, <i>Whatsapp</i> , <i>E-mail</i> e reuniões presenciais.	Trello, <i>Whatsapp</i> , <i>E-mail</i> , <i>Bitbucket</i> , <i>Skype</i> e reuniões presenciais.

Fonte: Própria autora.

Como pode ser observado através da análise do Quadro 29, nas duas experiências, houve a necessidade de adaptar o *Scrum* à realidade de cada equipe. Na conclusão do Guia do *Scrum* 2016, os seus autores manifestaram uma opinião sobre estas adaptações:

Papéis, artefatos, eventos e regras do *Scrum* são imutáveis e embora seja possível implementar somente partes do *Scrum*, o resultado não é *Scrum*. *Scrum* existe somente na sua totalidade, funcionando bem como um *container* para outras técnicas, metodologias e práticas. (SCHWABER; SUTHERLAND, 2016, p. 17).

O fato é que, mesmo não usando o *Scrum* em sua totalidade, projetos de desenvolvimento de *software* podem obter diversos benefícios aplicando os seus princípios, ainda que o adaptando. Exemplo disso são trabalhos como os de Simoyama e Bueno (2016) e Lima, Freire e Costa (2012) que corroboraram com esta afirmativa.

O primeiro utilizou o *Scrum* dentro de uma autarquia federal, obtendo resultados como ganho na agilidade das entregas dos seus projetos e sugerindo-o para organizações similares dentro da administração pública, que possuam problemas recorrentes com os seus cronogramas. Já o segundo, foi realizado em um Laboratório de Pesquisa e Desenvolvimento

de projetos de *software*, localizado dentro de uma Instituição de Ensino Superior do estado de Minas Gerais. Os projetos desenvolvidos visaram auxiliar a gestão ambiental de uma organização governamental deste mesmo estado. Os principais resultados da aplicação e adaptação do *framework Scrum* nesta pesquisa, apontaram melhorias no gerenciamento e no desenvolvimento dos seus projetos de *software*, bem como um maior envolvimento e comprometimento dos membros da equipe, permitindo um trabalho mais colaborativo.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma proposta de implantação e análise do *framework Scrum* no desenvolvimento de novas funcionalidades para o *website* Nosso Exercício. Buscando identificar quais benefícios e/ou dificuldades seriam encontrados com a aplicação deste método ágil neste ambiente, foram criadas estratégias rumo ao encontro destas respostas.

Inicialmente foi realizada uma análise do *website* à fim de se compreender o seu contexto e aplicação. Através de reuniões com os professores idealizadores do projeto foi criada uma lista de requisitos desejáveis para o *software* à fim de se compreender quais seriam as suas necessidades de evolução. Também foram identificadas a equipe de trabalho e as tecnologias envolvidas no seu desenvolvimento. Diante destas informações, devido à sua simplicidade, flexibilidade e buscando maior controle na execução do projeto, foi proposto o uso do *framework Scrum* para gerenciar as atividades de desenvolvimento do *software*.

O trabalho foi dividido em duas etapas. A Etapa 1 objetivou a capacitação de alunos do PET, integrantes do primeiro *Time Scrum* formado, no treinamento do próprio *Scrum* e das tecnologias de desenvolvimento utilizadas para o *software web*. Apesar dos avanços no conhecimento adquirido, esta equipe mostrou-se imatura para a continuidade do projeto diante dos limites de tempo disponíveis. Assim, na Etapa 2, um novo *Time Scrum* foi criado, mais experiente, objetivando o desenvolvimento das funcionalidades desejáveis para o *software*.

O método de pesquisa-ação foi o escolhido para descrever as duas etapas desta pesquisa. Inicialmente, fez-se um estudo preliminar do contexto geral do projeto, definindo o seu objetivo, as partes interessadas, o *status quo* do mesmo e as justificativas para o seu desenvolvimento. Em seguida, nos ciclos de condução, as ações para cada etapa foram efetivamente planejadas e executadas. Os resultados obtidos, foram posteriormente analisados durante a meta-fase. Foi apresentada uma analogia entre o ciclo da pesquisa-ação proposto por Coughlan e Coughlan (2002), mais especificamente o ciclo de condução, com o ciclo de desenvolvimento dos *Sprints* no *Scrum*.

Após a condução das duas etapas, através da análise dos resultados, foi possível elencar os seguintes benefícios obtidos com a implantação do *framework Scrum* neste projeto, a saber:

- O ritmo constante de execução dos *Sprints*, mantendo o foco do *Time* durante o seu trabalho, auxiliando no cumprimento das metas. A autora acredita que o foco poderia ser ainda maior se as equipes não precisassem estar envolvidas em outros projetos paralelos.

- A presença contínua do cliente (*Product Owner*) orientando o Time na realização de um trabalho em conformidade com as suas reais necessidades.

- A flexibilidade do *framework* em se ajustar às condições de ambiente e trabalho de cada Time criado, como já observado por Simoyama e Bueno (2016) e Lima, Freire e Costa (2012).

- O aprendizado constante do *software* desenvolvido, obtido através das discussões feitas nas Reuniões de Planejamento e Revisão.

- A possibilidade de realizar melhorias contínuas no processo utilizado, através das reflexões e cursos de ação definidos nas Reuniões de Retrospectiva.

- A motivação das equipes no uso do *Scrum*, mantendo-as comprometidas com os trabalhos realizados mesmo diante das dificuldades encontradas.

Mais especificamente no caso da primeira etapa realizada, o uso do *Scrum* proporcionou ao Time uma visão completa das atividades de desenvolvimento de *software*, dentro do tempo disponível para o projeto. Isto não seria possível, por exemplo, nas mesmas circunstâncias, com um método tradicional, em razão do rigor e nível de detalhamento exigidos pelo mesmo.

A primeira formação do Time de Desenvolvimento ainda pôde se beneficiar com um ganho de aprendizado relevante, obtido através do compartilhamento do conhecimento entre os membros. Este mesmo ganho foi mencionado na pesquisa de Rocha *et al.* (2017), devido a forma colaborativa de trabalho de um Time *Scrum*. Acredita-se que este poderia ser ainda mais evidenciado, se, este mesmo Time, tivesse tido a oportunidade de continuar o seu trabalho em novos ciclos *Scrum*.

No caso do segundo Time, composto por apenas um desenvolvedor, destaca-se como ganho o controle gerencial realizado, geralmente negligenciado em trabalhos semelhantes pela ausência de um processo leve que lhes propicie esta condição. (PAGOTTO, 2016).

O papel da *Scrum Master* também foi adaptado, pois, além das suas funções básicas, como remover impedimentos e ensinar o *Scrum* para o Time, a mesma se incumbiu de atividades como a criação dos critérios e testes de aceitação, realização de testes, controle das mudanças solicitadas pelo *Product Owner* e revisão constante do planejamento. Conforme relato do desenvolvedor, este trabalho contribuiu para que o mesmo mantivesse o foco nas atividades de desenvolvimento para o *software*, confrontando Sabbagh (2016, p .73) quando diz que “para times pequenos demais, é possível que o *Scrum* gere uma sobrecarga com suas regras, papéis e eventos que supere seus benefícios.”.

O ponto crítico deste trabalho deu-se sobre a imaturidade com as tecnologias utilizadas para o desenvolvimento do *website*. Em relação ao primeiro Time formado, a inexperiência dos alunos do PET, ainda em formação, permitiu que o *Scrum* fosse aplicado apenas a nível de capacitação. Mesmo obtendo ganhos de aprendizado no *framework* e nas tecnologias utilizadas, em decorrência do limite de tempo para a realização deste projeto, uma nova equipe foi considerada.

O segundo Time *Scrum* foi composto por membros com maior capacidade técnica para o desenvolvimento do trabalho, porém, deparou-se com obstáculos decorrentes da falta de experiência com o Drupal, ambiente no qual o Nosso Exercício foi desenvolvido. Apesar deste Time ter cumprido as suas metas em relação às funcionalidades solicitadas, esta deficiência implicou em atrasos nas entregas, impossibilidade de realizar estimativas e dificuldades na compreensão das complexidades das histórias usando a tecnologia mencionada.

Outra dificuldade observada pelo segundo Time formado foi lidar com as constantes alterações solicitadas pelo *Product Owner*. Gomes (2013) justifica, alertando que é importante estar preparado para as prováveis mudanças nos requisitos, uma vez que o cliente também está aprendendo e evoluindo suas ideias a respeito da solução que está sendo desenvolvida.

As lições aprendidas com esta pesquisa indicam que é possível obter benefícios com a implantação do *framework Scrum* que superem as suas dificuldades, desde que sejam feitas as devidas análises do ambiente em que o mesmo for adotado. No caso deste projeto, o critério de maior impacto a ser considerado foi o limite de tempo em que deveria ser aplicado. Para ambientes em que os critérios para implantação de um método de desenvolvimento de *software* forem mais rigorosos, recomenda-se uma análise mais criteriosa dos fatores que podem impactar negativamente no sucesso do projeto.

Como trabalho futuro, propõe-se o desenvolvimento das novas funcionalidades atualizadas no *Product Backlog*, garantindo a evolução do *website* Nosso Exercício, tornando-o cada vez mais completo. Além disso, seria ideal que, anteriormente ou paralelamente ao projeto, a equipe tivesse condições de se aperfeiçoar nas tecnologias demandadas para o mesmo, evitando usar do tempo de desenvolvimento para isso.

Ademais, sugere-se que, para os novos ciclos de desenvolvimento utilizando o *Scrum*, técnicas como o *Planning Poker*, gráfico de *Burdown* e estimativas sejam utilizadas com intuito de melhorar respectivamente, o nível de compreensão das funcionalidades, o controle sobre o ritmo do projeto e o esforço necessário para o desenvolvimento do trabalho a ser realizado.

REFERÊNCIAS

- ABT, Bill et al. **Manual PHP**. 2017. Disponível em: <https://secure.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 10/07/2017.
- BECK, K. et al. **Manifesto para o desenvolvimento ágil de *software***, 2001. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em 10/05/2016.
- BITBUCKET. O que é controle de versão. Disponível em: <<https://www.atlassian.com/git/tutorials/what-is-version-control>>. Acesso em: 03/02/2018.
- BROD, Cesar. **Scrum_ Guia prático para projetos ágeis**. 2ª Edição. São Paulo: Novatec, 2015. 198p.
- CAMPOS, Augusto. **O que é software livre**. BR-Linux. Florianópolis, março de 2006. Disponível em <<http://br-linux.org/linux/faq-softwarelivre>>. Acesso em 15/01/18.
- COHN, Mike. **User Stories Applied**. Addison-Wesley, 2004.
- COUGHLAN, Paul; COUGHLAN, David. **Action Research for Operation Management**. International Journal of Operations and Production Management, v.22, n. 2, p.220- 240, 2002. Disponível em: <<https://www.researchgate.net/publication/235278074>>. Acesso em 12/07/2017.
- CURTY, Pinheiro. **Guia Rápido para Gerenciar Projetos Scrum**, 18/05/2017. Disponível em: <http://imejunior.com.br/2017/05/18/guia-rapido-para-gerenciar-projetos-Scrum>. Acesso em: 12/07/2017.
- DICIONÁRIO INFOPÉDIA DA LÍNGUA PORTUGUESA**. Porto Editora, 2003-2018. Disponível em: <<https://www.infopedia.pt/dicionarios/lingua-portuguesa/código-fonte>>. Acesso em: 19/07/2017.
- ENGEL, Guido Irineu. **Pesquisa-Ação**. Educar, Curitiba, n. 16, p. 181-191. 2000. Editora da UFPR.
- FIALHO JR., Mozart. **Dicionário de Informática**. 2ª edição. Goiânia: Gráfica Terra Ltda. 2002, 190p.
- GABRIELI, Leandro. **Modelagem e Avaliação de um Sistema Modular para Gerenciamento de Informação na Web**. 137 f. Dissertação (Mestrado em Engenharia de Produção), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Disponível em: <<http://www.producao.ufrgs.br/arquivos/publicacoes/LeandroVettorazziGabrieli.pdf>> Acesso em: 10/02/18.
- GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 4ª edição. São Paulo: Atlas, 2002. 176p.
- GOMES, André Farias. **Agile: Desenvolvimento de *software* com entregas frequentes e foco no valor de negócio**. São Paulo: Casa do Código, 2013. 151p.

KURTZ, João. **O que é *software* de código aberto?** 04 mar. 2016. TechTudo. Disponível em: <<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2016/03/o-que-e-software-de-codigo-aberto.html>>. Acesso em: 10/02/2018.

LAMIM, Jonathan. **Afinal, o que é *Frontend* e o que é *Backend*?** Oficina da Net, 2014. Disponível em: <<https://www.oficinadanet.com.br/post/13541-afinal-o-que-e-frontend-e-o-que-e-backend->>. Acesso em: 20/02/2018.

LIMA, I.R.; FREIRE, T.C.; COSTA, H. A. X.. **Implantação e Adaptação do *Scrum* em um Laboratório de Pesquisa e Desenvolvimento de Projetos de *Software*.** Revista de Sistemas de Informação da FSMA nº9, 2012, p. 16-23. Disponível em: <http://www.fsma.edu.br/si/edicao9/FSMA_SI_2012_1_Principal_2.pdf>. Acesso em: 20/10/2017.

LUIZ, Rogério da Costa Dantas. **Plone 4: Administrando servidores 4.x na prática.** Rio de Janeiro: Brasport, 2014. 129p.

MARIOTTI, Flávio S. ***Kanban* o ágil adaptativo.** 45 ed. São Paulo. 2012. Disponível em: <<http://www.devmedia.com.br/kanban-o-agil-adaptativo-revista-engenharia-de-software-magazine-45/23560>>. Acesso em 05 de junho de 2017.

MELLO, Carlos Henrique Pereira et al. **Pesquisa-ação na engenharia de produção:** proposta de estruturação para sua condução. *Production*, v. 22, n. 1, p. 1-13, 2012.

MICROSOFT. **O que é o Skype?** Disponível em: <<https://support.skype.com/pt-br/faq/FA6/o-que-e-o-skype>>. Acesso em 04/12/17.

MIGUEL CAUCHICK, PAULO. **Aspectos relevantes no uso da pesquisa-ação na engenharia de produção.** *Exacta*, vol. 9, núm. 1, 2011, pp. 59-70. Universidade Nove de Julho, São Paulo, Brasil. Disponível em: <<http://www.redalyc.org/pdf/810/81018619006.pdf>>. Acesso em 13/09/2017.

NBR ISO/IEC 12207, (1998). **Tecnologia da Informação: processos de ciclo de vida de *software*,** ABNT, Rio de Janeiro.

PAGOTTO, Tiago *et al.* ***Scrum solo: Software process for individual development*,** 2016 *11th Iberian Conference on Information Systems and Technologies (CISTI)*. Gran Canaria, 2016, pp. 1-6. Disponível em: <URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7521555&isnumber=7521364>>. Acesso em 12/12/17.

PRESSMAN, Roger S. **Engenharia de *software*.** 6ª Edição. São Paulo: McGraw-Hill. 2006.720p.

PRESSMAN, Roger S. **Engenharia de *Software*:** uma abordagem profissional. 7ª Edição. São Paulo: McGraw Hill, 2011.779p.

PRODANOV, Cleber Cristiano. FREITAS, Ernani Cesar de. **Metodologia do trabalho científico [recurso eletrônico]: técnicas da pesquisa e do trabalho acadêmico.** 2ª edição. Novo Hamburgo: Feevale, 2013. Disponível em: <

<http://www.feevale.br/Comum/midias/8807f05a-14d0-4d5b-b1ad-1538f3aef538/E-book%20Metodologia%20do%20Trabalho%20Cientifico.pdf>>. Acesso em 12/06/2017.

ROCHA, Fabio Gomes *et al.* **O Scrum como forma de gerenciar alunos em projetos de desenvolvimento tecnológico e iniciação científica: um estudo de caso realizado na UNIT/SE.** Edapeci. São Cristóvão (SE), v.17, nº2, p.213-223, mai/ago 2017. Disponível em: < <https://seer.ufs.br/index.php/edapeci/article/view/6770/pdf>>. Acesso em:20/10/2017.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso.** Editora Casa do Código, 2ª Edição, 2016.

SCHWABER, Ken. SUTHERLAND, Jeff. **O Guia do Scrum. 2016.** Tradução: CRUZ (2016). Disponível em: < <http://www.fabiocruz.com.br/wp-content/uploads/2016/09/2016-Guia-doScrum-PtBR-v1FC.pdf>>. Acesso em: 12/12/2016.

SILVA, Giancarlo. **Gerencie equipes e tarefas com o Trello e dê adeus aos post-its!**, 2017. Disponível em: < <https://canaltech.com.br/utilitarios/gerencie-equipes-e-tarefas-com-o-trello-e-de-adeus-aos-post-its/>>. Acesso em: 12/07/2017.

SILVA, Maurício Samy. **Fundamento de HTML5 e CSS3.** São Paulo: Novatec, 2016.

SIMOYAMA, F. O.; BUENO, R. L. P. **Adaptação e implantação da metodologia Scrum para projetos ágeis numa Autarquia Federal.** Revista Gestão & Tecnologia, v. 16, n. 2, p. 260-276, 2016. Disponível em: <http://www.spell.org.br/documentos/ver/42768/adaptacao-e-implantacao-da-metodologia-scrum-para-projetos-ageis-numa-autarquia-federal->. Acesso em: 20/11/2017.

SOMMERVILLE, Ian. **Engenharia de Software.** 8ª Ed. Tradução Selma Shin Melnikoff; Reginaldo Arakaki; Edilson de Andrade Barbosa. São Paulo: Pearson, 2007.

SOUZA, Ailton. **Servidor WEB Apache, o que esperar?** 2017. DEVMEDIA. Disponível em: <<http://www.devmedia.com.br/servidor-web-apache-o-que-esperar/7096>>. Acesso em: 10/07/2017.

STAIR, Ralph M.; REYNOLDS, George W. **Princípios de Sistemas de Informação: uma abordagem gerencial.** Quarta Edição. Rio de Janeiro. LTC Editora, 2002. 496p.

STANDISH, Group. **The Standish Group Chaos Report, 2014.** Disponível em: < <http://blog.standishgroup.com/post/18> >. Acesso em:15 de junho de 2017.

STANDISH, Group. **About The Standish Group, 2017.** Disponível em: < <http://standishgroup.com/about> >. Acesso em: 28 de agosto de 2017.

SUTHERLAND, Jeff. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo.** Tradução: Natalie Gerhardt. São Paulo. LeYa, 2014.

TAKEUCHI, H.; NONAKA, I. **The New Product Development Game, Harvard Business Review**, p. 137-146, Jan-Feb. 1986.

TELES, Vinícius M. ***Extreme Programming – Aprenda Como Encantar Seus Usuários Desenvolvendo Software Com Agilidade E Alta Qualidade***. Rio De Janeiro. Novatec Editora, 2006. 316 p.

THIOLLENT, Michel. **Metodologia da Pesquisa-Ação**. 18ªEd. São Paulo: Cortez,2011.

TOMLINSON, Todd. **Desenvolvimento em Pro Drupal 7**. Terceira Edição. Rio de Janeiro. Ciência Moderna Ltda, 2012. 701p.

TRELLO. Disponível em: < <https://trello.com/>>. Acesso em: 10/12/2017.

TRIPP, David. **Pesquisa-ação: uma introdução metodológica**. Educação e Pesquisa, São Paulo, v. 31, n. 3, p. 443-466, set./dez. 2005.

VERGARA, Sylvia Constant. **Tipos de pesquisa em administração**. FGV/EBAP, Rio de Janeiro, nº52, junho 1990.

VERSIONONE. **11th Annual State of Agile Report**, 2016. Disponível em: < <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>> Acesso em: 06/07/2017.

VERSIONONE. **VersionOne: Pioneer em Agile Project Management & DevOps**, 2017. Disponível em: < <https://www.versionone.com/about/>> Acesso em: 30/08/2017.

APÊNDICE A – CICLO DE CONDUÇÃO DA ETAPA 2: *SPRINT 1*

Neste e nos próximos apêndices o leitor encontrará a demonstração de todas as histórias de usuário contempladas no ciclo de condução da Etapa 2 deste trabalho. Os apêndices foram divididos por *Sprints* e farão referência apenas às novas histórias consideradas em cada um. Assim, as histórias que entraram nos *Sprints* como pendências, não serão repetidas neste apêndice. Também não serão mencionadas as histórias que foram canceladas.

A história de ID8, referente a mudanças no *layout/usabilidade* do *site* não será mostrada isoladamente, pois, os seus resultados poderão ser conferidos nas telas apresentadas das demais histórias, após serem desenvolvidas, no decorrer das demonstrações feitas.

O Quadro 30 foi criado para situar o leitor numa visão geral de quais histórias foram iniciadas em cada *Sprint*, durante a Etapa 2.

Quadro 30 - Histórias desenvolvidas em cada *Sprint* da Etapa 2

<i>Product Backlog - Projeto Nosso Exercício</i>		
ID	História	<i>Sprint</i>
1	Como usuário autenticado, quero ter a opção de realizar <i>download</i> dos exercícios e/ou listas de exercícios nos formatos .pdf (<i>Adobe</i>), .odt (<i>Writer-Libre Office</i>), .tex (<i>Latex</i>), e .doc (<i>Word</i>) para que os exercícios já venham formatados como estão no <i>site</i> .	2
2	Como usuário autenticado, ao criar uma lista, quero escolher a quantidade de exercícios que devem compor esta lista.	4
3	Como usuário autenticado, quero ter a opção de resolver questões de múltipla escolha no sistema e obter o resultado para que eu possa treinar os meus estudos.	5
4	CANCELADA	-
5	CANCELADA	-
6	Como usuário autenticado, navegando em uma lista pública, desejo ter a opção visualizar completamente as informações de cada exercício da lista.	2
7	Como usuário autenticado ou registrado, desejo visualizar na tela inicial, sugestões de disciplinas já cadastradas no <i>site</i> e realizar a busca por exercícios, clicando em uma destas opções, para que eu possa ter mais de uma forma de realizar uma busca no sistema.	1
8	Como <i>Product Owner</i> desejo obter melhorias no <i>layout/usabilidade</i> do <i>site</i> para proporcionar ao usuário uma melhor experiência de navegação.	7
9	Como usuário anônimo ou autenticado, ao realizar uma busca e inserir um termo e/ou disciplina, desejo que o <i>software</i> retorne apenas os exercícios ‘com resolução’ e/ou ‘comentários’, para que eu possa criar uma lista ou consultar os exercícios que atendam a estes critérios.	1
10	Como usuário anônimo ou autenticado, ao realizar uma pesquisa, quero que o sistema retorne a quantidade de itens resultantes da busca.	5
11	CANCELADA	-
12	Como administrador, quando o usuário registrado clicar nos <i>links</i> “Denunciar erro” e/ou “Denunciar infração”, quero que o sistema envie uma mensagem para que o usuário confirme a sua ação, gerando maior confiabilidade à denúncia.	4

Para cada *Sprint* são apresentadas as descrições das histórias trabalhadas, as justificativas para as mesmas, os critérios e teste de aceitação criados, bem como as alterações no *layout* do *website* Nosso Exercício, após o desenvolvimento destas histórias.

No *Sprint 1* foram contempladas as histórias de ID7 e ID9 como demonstradas a seguir. O Quadro 31 apresenta a descrição da história 7.

Quadro 31 - Descrição da História 7 (Sprint1- Etapa2)

ID da História: 7	Prioridade: Alta	Sprint: 1
Nome da História: Mostrar opções de disciplinas existentes ao realizar uma busca		
Descrição:		
Como usuário anônimo ou registrado, desejo visualizar na tela inicial, sugestões de disciplinas já cadastradas no <i>site</i> e realizar a busca por exercícios, clicando em uma destas opções, para que eu possa ter mais de uma forma de realizar uma busca no sistema.		
Pontos estimados:		Pontos consumidos:

Justificativa: a tela inicial do Nosso Exercício na sua versão original, disponibilizava apenas uma opção para o usuário que desejasse fazer uma pesquisa. Entendendo que desta forma o sistema contribuía apenas com aqueles que já sabiam o termo pelo qual deveriam pesquisar, o *Product Owner*, solicitou uma nova opção de pesquisa, mais intuitiva, que sugerisse para o usuário as opções de disciplinas já disponíveis.

Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 32 e 33.

Quadro 32 - Critério de Aceitação- História7 (Sprint 1- Etapa2)

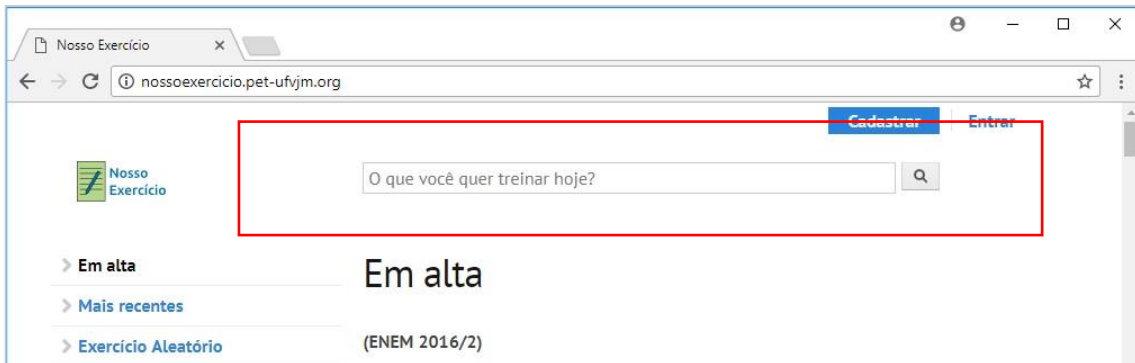
Critérios de Aceitação	
História:	7
Critério 1: As pesquisas no <i>site</i> deverão ser realizadas digitando o termo diretamente no campo de busca, ou, selecionando um dos <i>links</i> com os nomes das disciplinas que ficarão disponíveis abaixo do campo de busca.	

Quadro 33 - Teste de Aceitação- História7 (Sprint 1- Etapa2)

Teste de Aceitação		
História:	7	
Número	Ação	Resultado Esperado
1	- O usuário digita um termo no campo de busca.	- O sistema retorna todos os exercícios que se relacionam com o termo digitado.
2	- O usuário clica no <i>link</i> do nome de um disciplina.	- O sistema retorna todos os exercícios que se relacionam com o <i>link</i> selecionado.

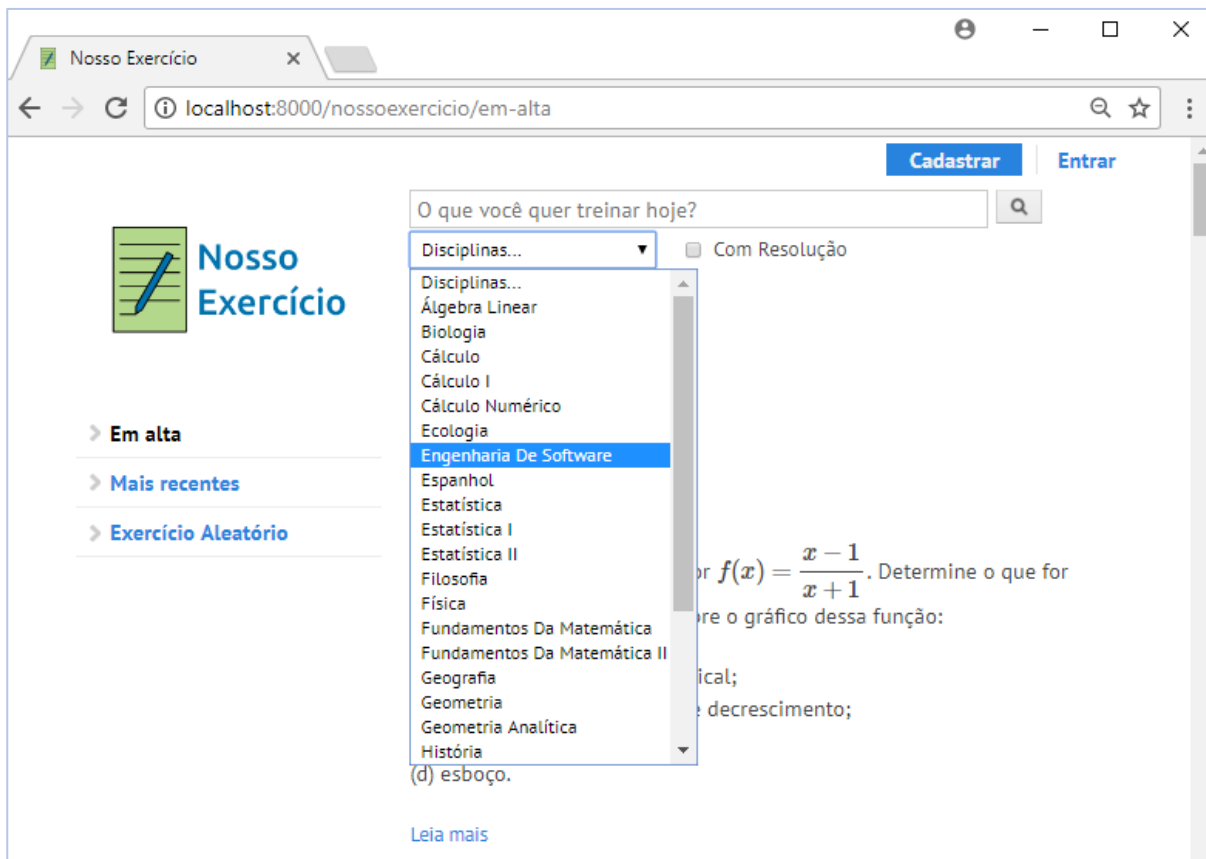
A Figura 22 demonstra o *layout* do Nosso Exercício antes do desenvolvimento da história 7. O retângulo em vermelho destaca a barra de busca do *site*.

Figura 22 - Layout do Nosso Exercício antes do desenvolvimento da história 7



A figura 23 mostra como ficou o *layout* do *website* após o desenvolvimento da história 7.

Figura 23 – Layout do Nosso Exercício após o desenvolvimento da história 7



O Quadro 34 apresenta a descrição da história de usuário 9.

Quadro 34 - Descrição da História 9 (Sprint1- Etapa2)

ID da História: 9	Prioridade: Alta	Sprint: 1
Nome da História:	Realizar buscas apenas por exercícios que contenham respostas e/ou comentários	
Descrição:		
Como usuário anônimo ou registrado, ao realizar uma busca e inserir um termo e/ou disciplina, desejo que o <i>software</i> retorne apenas os exercícios ‘com resolução’ e/ou ‘comentários’, para que eu possa criar uma lista ou consultar os exercícios que atendam a estes critérios.		
Pontos estimados:		Pontos consumidos:

Justificativa: Esta história visou oferecer ao usuário do sistema uma opção de filtro na realização de buscas por exercícios ‘com resolução’ e/ou ‘com comentários’. Antes da implementação da mesma, a busca retornava todos os exercícios cadastrados, não permitindo ao usuário interferir neste processo. Após a *Sprint Review* o *Product Owner* decidiu manter apenas a opção de buscas de exercícios ‘com resolução’.

Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 35 e 36.

Quadro 35 - Critério de Aceitação- História 9 (Sprint 1- Etapa2)

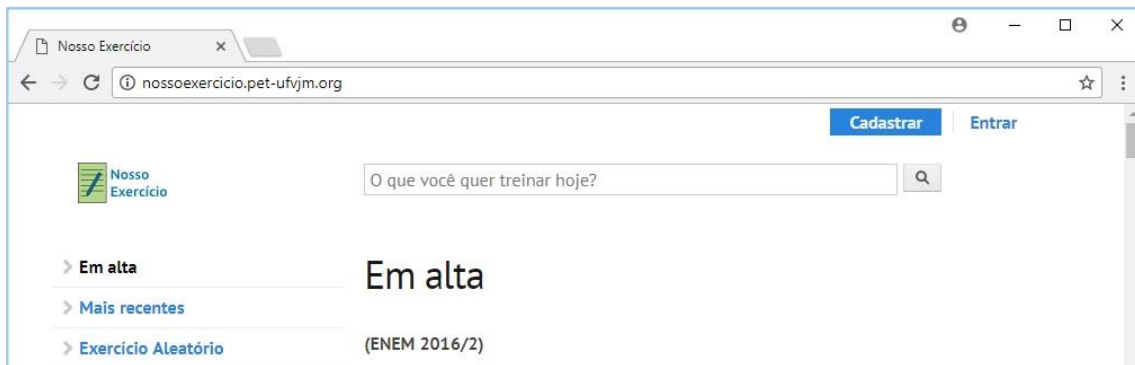
Critério(s) de Aceitação	
História:	9
Critério 1: Na página inicial, ao realizar uma busca, ao habilitar o campo ‘com resolução’ e/ou ‘com comentários’ e confirmar esta busca, o sistema deverá mostrar apenas exercícios que contenham resposta e/ou comentários.	

Quadro 36 - Teste de Aceitação- História 9 (Sprint 1- Etapa2)

Teste de Aceitação		
História:	9	
Número	Ação	Resultado Esperado
1	- O usuário digita um termo no campo de busca e seleciona a opção ‘Com resolução’. - O usuário clica no ícone da lupa ou em um <i>link</i> de uma disciplina para confirmar a busca.	- O sistema retorna todos os exercícios que se relacionam com o termo digitado, que contenham resolução .

A figura 24 demonstra a tela inicial do Nosso Exercício antes do desenvolvimento da história 9.

Figura 24 - Layout do Nosso Exercício antes do desenvolvimento da história 9



A opção 'Com resolução', desenvolvida na história 7 é demonstrada na Figura 25 (destaque em vermelho).

Figura 25 - Layout do Nosso Exercício depois do desenvolvimento da história 9



APÊNDICE B – CICLO DE CONDUÇÃO DA ETAPA 2: *SPRINT 2*

Neste *Sprint* as novas histórias consideradas foram as de ID6 e ID1. A descrição da história 6 é apresentada no Quadro 37.

Quadro 37 -Descrição da História 6 (*Sprint2- Etapa2*)

ID da História: 6	Prioridade: Alta	Sprint: 2
Nome da História:	Em listas públicas, exibir <i>link</i> para visualização completa do exercício.	
Descrição:		
Como usuário registrado, navegando em uma lista pública, desejo ter a opção visualizar completamente as informações de cada exercício da lista.		
Pontos estimados:		Pontos consumidos:

Justificativa: Na versão anterior do Nosso Exercício, quando uma lista pública fosse acessada por um usuário anônimo ou autenticado (que não fosse o criador da lista), era possível visualizar os exercícios apenas dentro da lista, não tendo acesso a todas as suas informações. Esta história visou possibilitar a estes usuários (e não apenas aos administradores/criadores de uma lista pública) a visualização dos exercícios dentro da lista e, caso necessário, fora da lista, tendo acesso assim a todas as informações pertencentes a estes exercícios.

Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 38 e 39.

Quadro 38 - Critério de Aceitação- História 6 (*Sprint 2- Etapa2*)

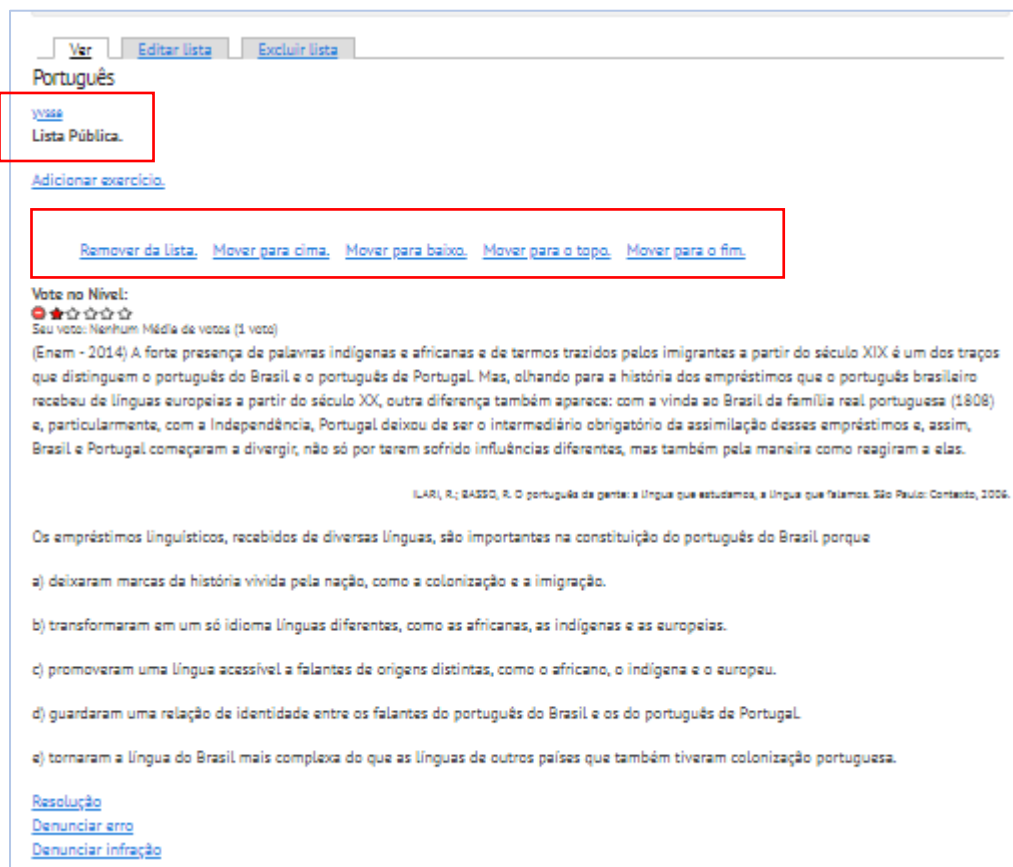
Critério(s) de Aceitação	
História:	6
Critério 1: Ao clicar no <i>link</i> de um exercício de uma lista pública, o sistema deverá exibir o exercício com a sua descrição completa.	

Quadro 39 - Teste de Aceitação- História 6 (Sprint 2- Etapa2)

Teste de Aceitação		
História:	6	
Número	Ação	Resultado Esperado
1	- O usuário está navegando em uma lista pública e clica no <i>link</i> 'Ver Exercício'.	- O sistema abre a página que contém a descrição completa deste exercício com a opção de voltar para a lista.
2	- O usuário visualiza o exercício e clica na opção de voltar para a lista.	- O sistema direciona o usuário novamente para a lista.

Através da Figura 26 é possível identificar um trecho de uma lista pública com as opções de visualização para os exercícios desta lista, antes do desenvolvimento desta funcionalidade.

Figura 26 - Layout do Nosso Exercício antes do desenvolvimento da história 6



Foi acrescentado o *link* 'Ver Exercício' que permite ao usuário abrir em uma nova página cada exercício isoladamente, visualizando todas as suas informações (Figura 27).

Figura 27 - *Layout* do Nosso Exercício após o do desenvolvimento da história 6

Português

[nossoexercicio](#)
Lista Pública.


[Adicionar exercício.](#)

Download da Lista

Corrigir Simulado

[Remover da lista.](#) [Mover para cima.](#) [Mover para baixo.](#) [Mover para o topo.](#) [Mover para o fim.](#)

[Ver Exercício](#)

Vote no Nível:

 Seu voto: Nenhum Média de votos (1 voto)
 (ENEM 2015)

Dia do Músico, do Professor, da Secretária, do Veterinário... Muitas são as datas comemoradas ao longo do ano e elas, ao darem visibilidade a segmentos específicos da sociedade, oportunizam uma reflexão sobre a responsabilidade social desses segmentos. Nesse contexto, está inserida a propaganda da Associação Brasileira de Imprensa (ABI), em que se combinam elementos verbais e não verbais para se abordar a estreita relação entre imprensa, cidadania, informação e opinião. Sobre essa relação, depreende-se do texto da ABI que,

a) para a imprensa exercer seu papel social, ela deve transformar opinião em informação.

b) para a imprensa democratizar a opinião, ela deve selecionar a informação.

c) para o cidadão expressar sua opinião, ele deve democratizar a informação.

d) para a imprensa gerar informação, ela deve fundamentar-se em opinião.

O Quadro 40 apresenta a descrição da história de usuário 1.

Quadro 40 - Descrição da História 1 (*Sprint2- Etapa2*)

ID da História:	1	Prioridade:	Alta	Sprint:	2
Nome da História:	Realizar <i>download</i> das listas de exercícios em diversos formatos				
Descrição:	Como usuário registrado , quero ter a opção de realizar <i>download</i> das listas de exercícios nos formatos .pdf (<i>Adobe</i>), .odt (<i>Writer-Libre Office</i>), .tex (<i>Latex</i>), e .doc (<i>Word</i>) para que os exercícios já venham formatados como estão no <i>site</i> .				
Pontos estimados:		Pontos consumidos:			

Justificativa: Para o usuário que quisesse realizar o *download* de uma ou mais listas de exercícios, era necessário usar o recurso clássico ‘Copiar e Colar’, e, ainda assim, ao fazê-lo,

recebia boa parte dos exercícios totalmente desconfigurados, principalmente aqueles envolvendo fórmulas matemáticas. Para esta funcionalidade o *Product Owner* solicitou que o Nosso Exercício disponibilizasse facilmente ao usuário, várias opções de formato para a realização do *download* das listas de exercícios. Durante o desenvolvimento desta história o Time compreendeu que a sua implementação seria mais complexa do que prevista e, em conformidade com o *Product Owner*, neste trabalho, foi disponibilizada apenas a opção de *downloads* em formato PDF.

Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 41 e 42.

Quadro 41- Critério de Aceitação- História 1 (Sprint 2- Etapa2)

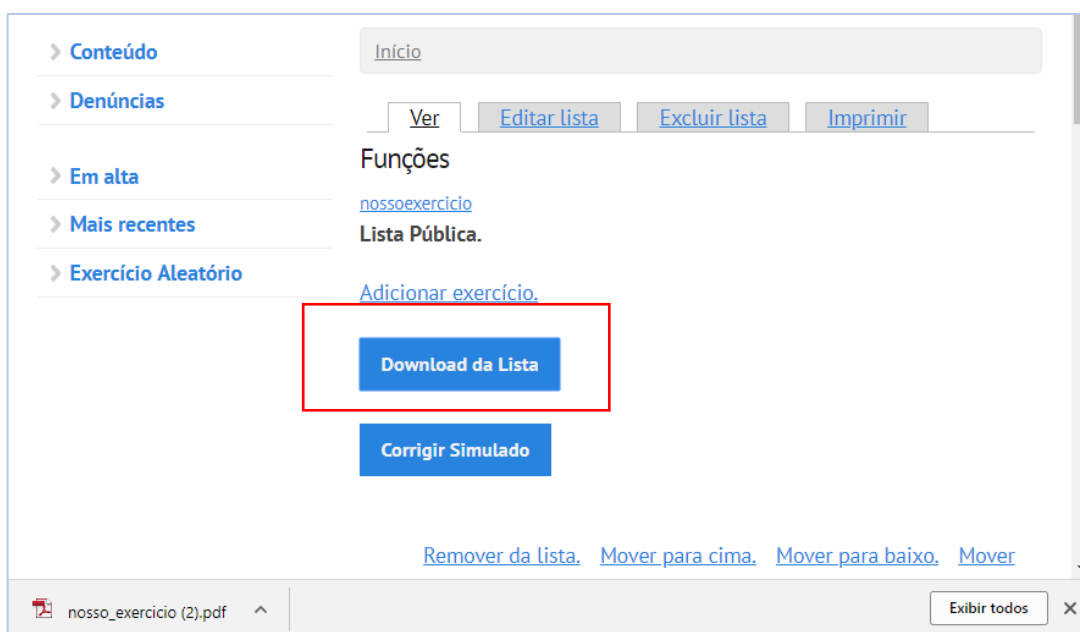
Critério(s) de Aceitação	
História:	1
<p>Critério 1: Ao visualizar um exercício e/ou lista de exercícios, o usuário logado no sistema, terá como opção clicar no botão ‘Download da lista’, e ao clicar sobre o mesmo, realizar o <i>download</i> dos exercícios daquela lista.</p>	

Quadro 42 - Teste de Aceitação- História 1 (Sprint 2- Etapa2)

Teste de Aceitação		
História:	1	
Número	Ação	Resultado Esperado
1	- O usuário visualiza uma lista de exercícios e clica no botão ‘Download da lista’	- O sistema disponibiliza a lista de exercícios através da barra de <i>status</i> do navegador.
2	- O usuário clica no arquivo na barra de <i>status</i> .	- O sistema abre uma nova aba no navegador com a lista de exercício em formato PDF para ser visualizada, salva ou impressa.

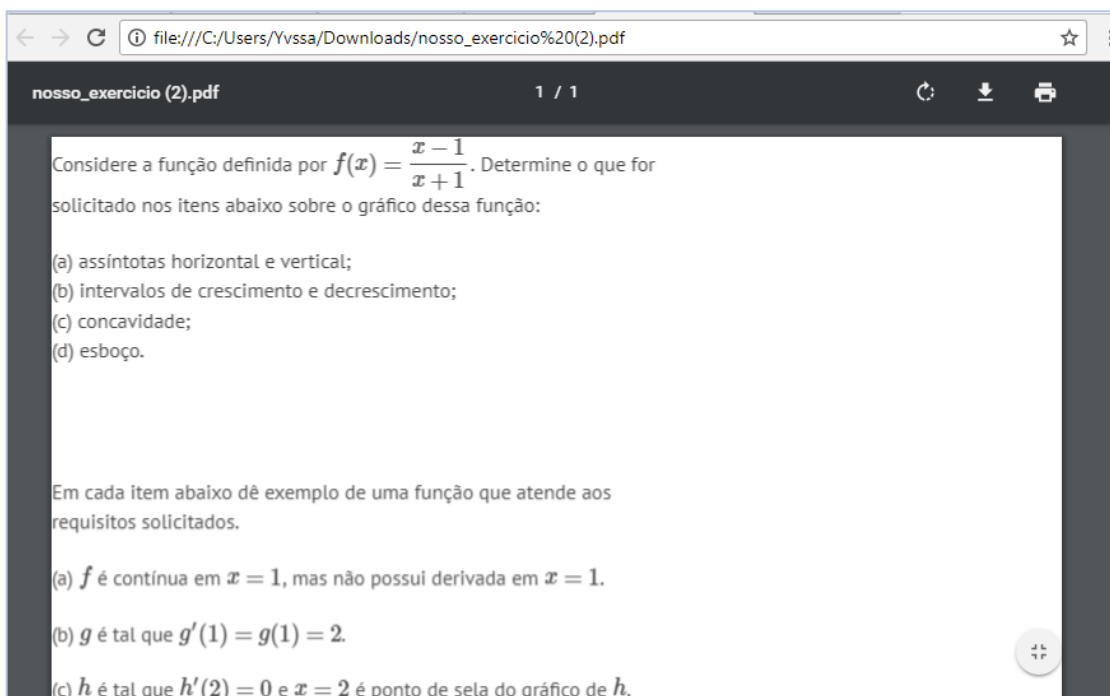
Como esta é uma nova funcionalidade no sistema não é possível mostrar o *layout* anterior à mesma. Assim, a Figura 28 mostra a tela do Nosso Exercício após o desenvolvimento da história 1, com o botão ‘Download da Lista’ para o usuário que desejar baixar uma lista de exercícios em PDF. No exemplo, o nome da lista é ‘Funções’ e por ser pública, está disponível para qualquer usuário a visualizá-la ou baixá-la.

Figura 28 - Botão para realizar *download* de uma lista em PDF



A Figura 29 mostra a aba que é aberta no navegador quando o usuário clica na lista após o seu *download*. Através desta tela o usuário pode salvar a lista, imprimi-la ou visualizá-la.

Figura 29 - Lista de exercícios disponível para baixar ou imprimir



APÊNDICE C – CICLO DE CONDUÇÃO DA ETAPA 2: *SPRINT 4*

Como o *Sprint 3* foi destinado apenas para pendências dos *Sprints* anteriores, não contemplando novas histórias, o mesmo não foi considerado neste apêndice. As histórias do *Sprint 4* foram as de ID2 e ID12 e serão demonstradas a seguir.

O Quadro 43 apresenta a descrição da história 2.

Quadro 43 - Descrição da História 2 (*Sprint4- Etapa2*)

ID da História: 2	Prioridade: Alta	Sprint: 4
Nome da História:	Criar lista com quantidade ‘n’ de exercícios escolhidos pelo usuário.	
Descrição:		
Como usuário autenticado, ao criar uma lista, quero escolher a quantidade de exercícios que devem compor esta lista.		
Pontos estimados:		Pontos consumidos:

Justificativa: Antes desta funcionalidade, ao criar uma lista de exercícios o usuário não tinha a opção de escolher a quantidade de exercícios que gostaria, para serem adicionados na mesma. Desta forma, todos os exercícios que atendiam ao termo pesquisado eram retornados na busca e o usuário deveria percorrer pelas páginas, adicionando os exercícios que queria.

Através desta história, o *Product Owner* solicitou que o sistema apresentasse em uma nova aba ‘Exercícios Aleatórios’, um campo para inserir o termo pesquisado, bem como a quantidade de exercícios desejados para a busca. Após o retorno da busca, o usuário poderia então adicionar os exercícios na lista. Caso não haja no banco de dados a quantidade de exercícios solicitados pelo usuário, o sistema retorna a quantidade encontrada.

Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 44 e 45.

Quadro 44 - Critério de Aceitação- História 2 (*Sprint 4- Etapa2*)

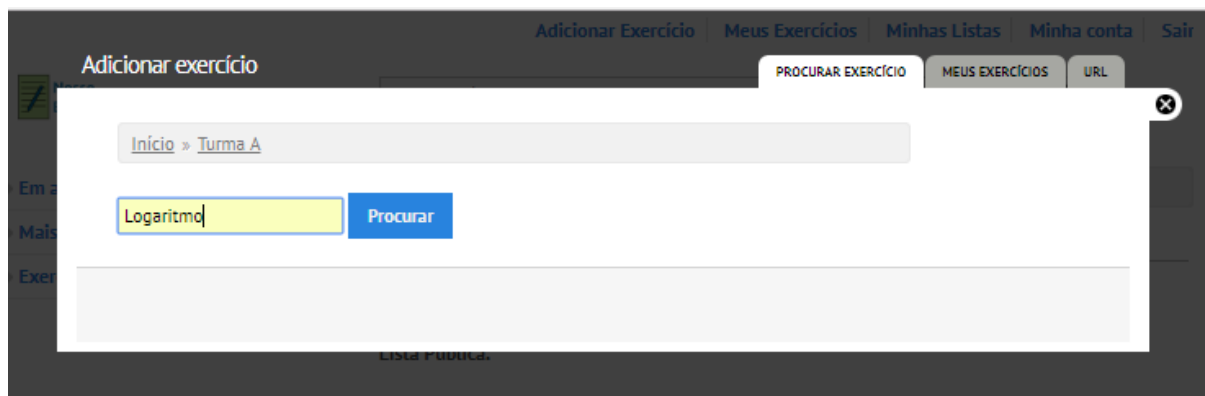
Critério(s) de Aceitação	
História:	2
Critério 1: Ao criar uma lista, o usuário terá a opção de determinar a quantidade de exercícios que serão retornados sobre um determinado assunto. Ao clicar num botão de confirmação a lista deverá ser criada com a quantidade de exercícios desejados.	

Quadro 45 - Teste de Aceitação- História 2 (Sprint 4- Etapa2)

Teste de Aceitação		
História:	2	
Número	Ação	Resultado Esperado
1	- O usuário está logado e clica em Minhas Listas/Adicionar Lista	- O sistema abre a página com os campos Título da Nova Lista, Visibilidade e Tipo.
2	- O usuário preenche os dados da nova lista e clica em Adicionar.	- O sistema abre a página da nova lista e exibe um <i>link</i> para o usuário adicionar exercícios na lista.
3	- O usuário clica no <i>link</i> .	- O sistema abre uma nova janela para que o usuário insira a quantidade de exercícios que deseja para a lista e o termo pesquisado.
4	- O usuário insere os dados e clica em confirmar.	- O sistema abre uma página da lista gerada com as opções: 'Aceitar Lista' e 'Gerar novamente'.
5	- O usuário clica na opção 'Gerar novamente'.	- O sistema abre uma nova janela para que o usuário insira novamente a quantidade de exercícios que deseja para a lista e o assunto.
6	- O usuário insere os dados e clica em confirmar.	- O sistema abre uma página com a lista gerada com as opções: 'Aceitar Lista' e 'Gerar novamente'.
7	- O usuário clica na opção 'Aceitar Lista'.	- O sistema insere os exercícios na lista criada.

A Figura 30 mostra que antes do desenvolvimento da história 2 a opção de 'Adicionar exercício' numa lista não permitia inserir a quantidade de exercícios desejados para a busca, apenas o termo.

Figura 30 - Adicionar exercício antes do desenvolvimento da história 2



Após o desenvolvimento da história 2, conforme mostrado na Figura 31, o usuário vê as opções para a criação de uma nova lista no sistema. Depois de clicar em 'Adicionar' o usuário é direcionado para outra tela em que insere um termo de busca e a quantidade de exercícios que deseja que sejam retornados desta busca, no exemplo foram 10 exercícios.

Figura 31 – Tela do *site* com as opções para criar uma nova lista

The figure consists of two overlapping screenshots of a web application interface. The left screenshot shows a form titled 'Adicionar lista.' with a breadcrumb 'Início » nossoexercicio'. The form has three sections: 'Título da nova Lista:' with a text input containing 'Turma A'; 'Visibilidade:' with a dropdown menu set to 'Pública'; and 'Tipo:' with a dropdown menu set to 'Lista'. A blue 'Adicionar' button is at the bottom. The right screenshot shows a search results page with a breadcrumb 'Início » Turma A'. It features a search bar with 'Logar' and a 'Quantidade de Exercícios' field with the value '10'. A blue 'Procurar' button is next to the search bar.

A Figura 32 mostra apenas um trecho dos exercícios retornados através da busca realizada para criar uma nova lista. Atenção para a caixa de seleção ‘Adicionar na lista’ que permite ao usuário escolher os exercícios para a lista criada. Após esta confirmação é possível ‘Aceitar Lista’ ou ‘Gerar Novamente’.

Figura 32 - Trecho de uma lista criada com as opções de 'Gerar Novamente' e 'Aceitar Lista'

The figure shows a snippet of a list item. At the top, there is a checkbox labeled 'Adicionar na lista.' which is checked. Below this is a text description of a math problem: 'Suponha que um motorista não pode dirigir se o nível de álcool no seu sangue for superior a 0,2 g por litro. Considere que o nível N de álcool por litro de sangue de um homem adulto, em gramas, decresça de acordo com a função $N(t) = N_0 \left(\frac{1}{2}\right)^t$, onde t representa o tempo, em horas, e N_0 representa o nível inicial de álcool por litro de sangue. Certo homem, adulto, ingeriu grande quantidade de bebida alcoólica e o nível de álcool em seu sangue chegou a 2 g por litro. Quantas horas e minutos ele terá que esperar para poder dirigir? (Dado: considere que $\log 2 \approx 0,3$).'. Below the text, there is a link 'Leia mais | 2 leituras' and two blue buttons: 'Gerar Novamente' and 'Aceitar Lista'.

O Quadro 46 apresenta a descrição da história 12.

Quadro 46 - Descrição da História 12 (Sprint 4- Etapa2)

ID da História:	12	Prioridade:	Alta	Sprint:	4
Nome da História:	Enviar mensagem de confirmação ao clicar em “Denunciar erro” e/ou “Denunciar infração”				
Descrição:	<p>Como administrador, quando o usuário autenticado clicar nos <i>links</i> “Denunciar erro” e/ou “Denunciar infração”, quero que o sistema faça com que usuário confirme a sua ação, gerando maior confiabilidade à denúncia.</p>				
Pontos estimados:		Pontos consumidos:			

Justificativa: Da forma que estava o Nosso Exercício antes do desenvolvimento desta história, com apenas um clique na opção ‘Denunciar erro’, o erro já era enviado para análise, e/ou, no caso de “Denunciar infração”, já despublicava o conteúdo. O *Product Owner* solicitou que fosse criado algum mecanismo de confirmação, para que o usuário autenticado pudesse confirmar a sua ação, evitando denúncias decorrentes de cliques não intencionais, por exemplo. Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 47 e 48.

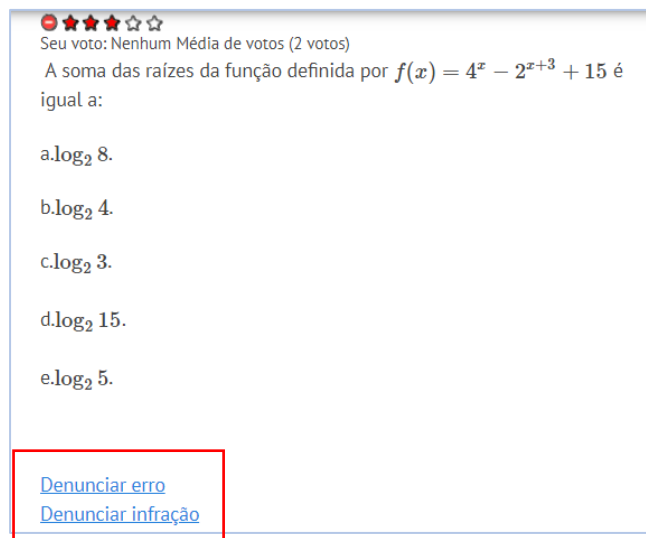
Quadro 47 - Critério de Aceitação- História 12 (Sprint 4- Etapa2)

Critério(s) de Aceitação	
História:	12
<p>Critério 1: Ao clicar em “Denunciar erro” e/ou “Denunciar infração” o usuário terá a opção de clicar em botão para que ele confirme a ação desejada.</p>	

Quadro 48 - Teste de Aceitação- História 12 (Sprint 4- Etapa2)

Teste de Aceitação		
História:	12	
Número	Ação	Resultado Esperado
1	- O usuário está logado e clica em “Denunciar erro” .	- O sistema abre a página mostrando um botão “Denunciar erro” .
2	- O usuário clica no botão “Denunciar erro” .	- O sistema envia a mensagem de confirmação da denúncia.
3	- O usuário está logado e clica em “Denunciar infração” .	- O sistema abre a página mostrando um botão “Denunciar infração”
4	- O usuário clica no botão “Denunciar infração” .	- O sistema envia a mensagem de confirmação da denúncia

Na Figura 33 é possível visualizar os *links* ‘Denunciar erro’ e ‘Denunciar infração’ abaixo da descrição de um exercício.

Figura 33 - Exercício com as opções 'Denunciar erro' e 'Denunciar infração'

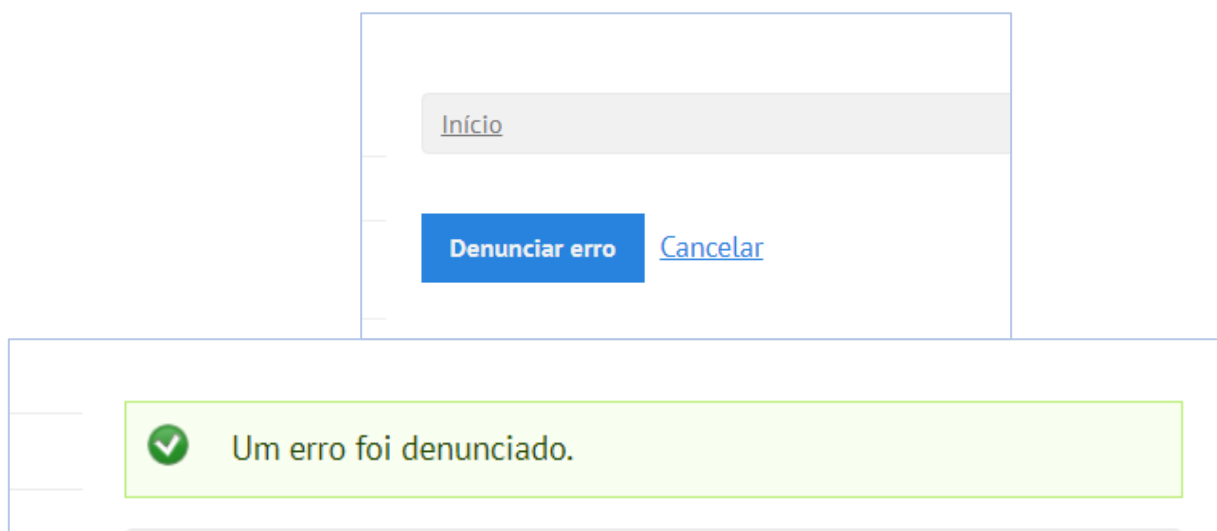
Seu voto: Nenhum Média de votos (2 votos)

A soma das raízes da função definida por $f(x) = 4^x - 2^{x+3} + 15$ é igual a:

- a. $\log_2 8$.
- b. $\log_2 4$.
- c. $\log_2 3$.
- d. $\log_2 15$.
- e. $\log_2 5$.

[Denunciar erro](#)
[Denunciar infração](#)

Ao clicar na opção 'Denunciar erro' por exemplo, o sistema abre uma página para que o usuário confirme a denúncia clicando no botão 'Denunciar erro'. Após o clique, o sistema mostra uma mensagem de confirmação da denúncia. Passos similares podem ser realizados para a opção 'Denunciar infração'.

Figura 34 - Telas para confirmação da opção 'Denunciar erro'

The top screenshot shows a confirmation dialog with a grey 'Início' button at the top, a blue 'Denunciar erro' button, and a blue 'Cancelar' link.

The bottom screenshot shows a green confirmation message box with a checkmark icon and the text 'Um erro foi denunciado.'

APÊNDICE D – CICLO DE CONDUÇÃO DA ETAPA 2: *SPRINT 5*

As histórias implementadas neste *Sprint* foram as de ID10 e ID3. O Quadro 49 apresenta a descrição da história 10.

Quadro 49 - Descrição da história 10 (*Sprint 5- Etapa 2*)

ID da História:	10	Prioridade:	Alta	Sprint:	5
Nome da História:	Mostrar quantidade de itens resultantes da busca.				
Descrição:	<p>Como usuário anônimo ou autenticado, ao realizar uma pesquisa, quero que o sistema retorne a quantidade de itens resultantes da busca.</p>				
Pontos estimados:		Pontos consumidos:			

Justificativa: na versão original do Nosso Exercício ao realizar uma busca através da barra de pesquisa, o sistema retornava a busca conforme a solicitação do usuário, porém, não era possível visualizar a quantidade de itens resultantes daquela busca. Desta forma, para saber quantos exercícios haviam sido retornados, o usuário deveria percorrer por todas as páginas para obter esta informação.

Os critérios e testes de aceitação criados para esta história são mostrados nos Quadros 50 e 51.

Quadro 50 - Critério de Aceitação para a história de usuário 10

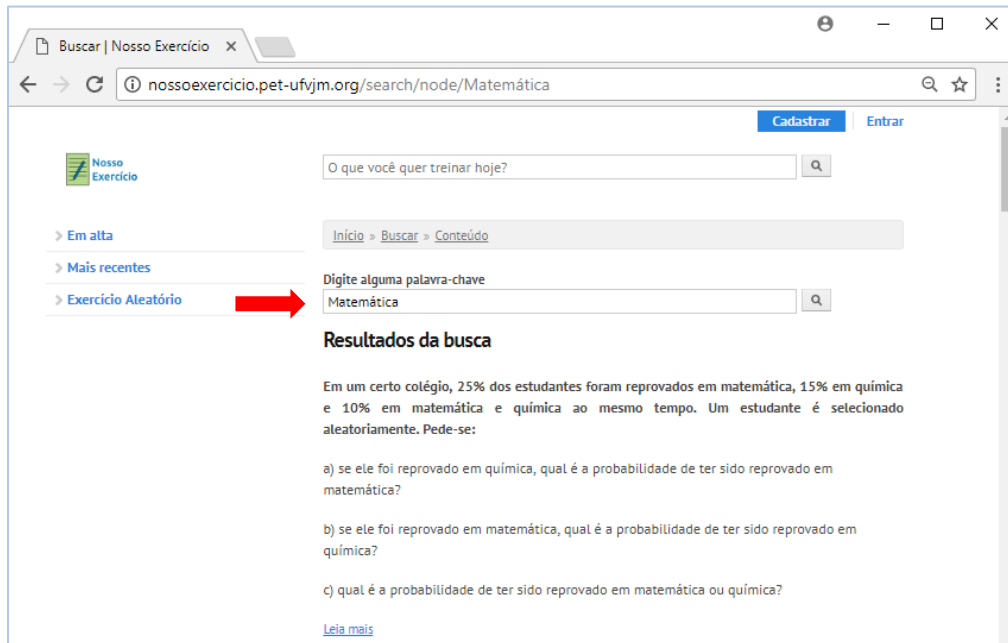
Critérios de Aceitação	
História:	10
Critério 1: A quantidade de itens da busca deverá aparecer após o texto ‘Resultados da busca’ seguindo o formato: “Página X de Y exercícios encontrados”, devendo X atualizar quando houver mais de uma página.	

Quadro 51 - Testes de Aceitação para a história de usuário 10

Teste de Aceitação		
História:	10	
Número	Ação	Resultado Esperado
1	- O usuário está logado, na página inicial do <i>site</i> e realiza uma pesquisa digitando um termo no campo de busca.	- O sistema abre uma página com o retorno da busca, mostrando a quantidade de itens encontrados e o número da página resultante daquela busca que o usuário se encontra. (Ex: Página 1 de 34 exercícios encontrados).
2	- O usuário clica em próxima página.	- O sistema abre uma página atualizando o número da página resultante da busca e mostra novamente o total de itens encontrados na busca anterior. (Ex: Página 2 de 34 exercícios encontrados).

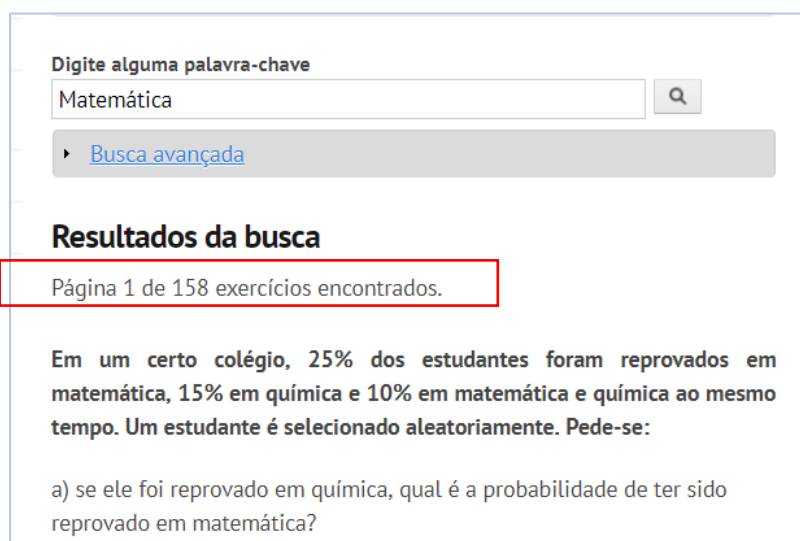
A Figura 35 apresenta uma busca com o termo ‘Matemática’, indicado pela seta vermelha, antes do desenvolvimento desta história.

Figura 35 - Busca pelo termo Matemática antes do desenvolvimento da história 10



Já na Figura 36, após o desenvolvimento da história é possível identificar que, realizando a mesma busca, o sistema agora retorna a página e a quantidade de exercícios encontrados com o termo pesquisado (realçados em vermelho).

Figura 36 - Busca pelo termo Matemática após o desenvolvimento da história 10



Na sequência deste *Sprint* a história de ID3 foi implementada. O Quadro 52 apresenta a descrição da história 3.

Quadro 52 - Descrição da história 3 (Sprint 5- Etapa 2)

ID da História:	3	Prioridade:	Alta	Sprint:	5
Nome da História:	Criar questões de múltipla escolha.				
Descrição:	<p>Como usuário autenticado, quero ter a opção de resolver questões de múltipla escolha no sistema e obter o resultado para que eu possa treinar os meus estudos.</p>				
Pontos estimados:		Pontos consumidos:			

Justificativa: esta é uma nova funcionalidade que permite a criação de um simulado no sistema, possibilitando ao usuário a resolução das questões e obtenção do resultado do seu treinamento. Os critérios de aceitação criados para esta história são mostrados nos Quadros 53

Quadro 53- Critério de Aceitação para a história de usuário 3

Critérios de Aceitação	
História:	3
Critério 1:	O campo 'Gabarito' deverá ser criado para que, ao adicionar um novo exercício com questões do tipo múltipla escolha, a resposta desta questão seja guardada neste campo.
Critério 2:	Um novo <i>link</i> 'Gerar Simulado' deverá ser criado na página inicial para que o usuário, quando estiver logado, possa criar uma lista com questões que farão parte do simulado. Ao criar a lista o sistema deverá retornar apenas questões do tipo múltipla escolha.
Critério 3:	O sistema deverá retornar o total de questões corretas e incorretas respondidas pelo usuário no simulado.

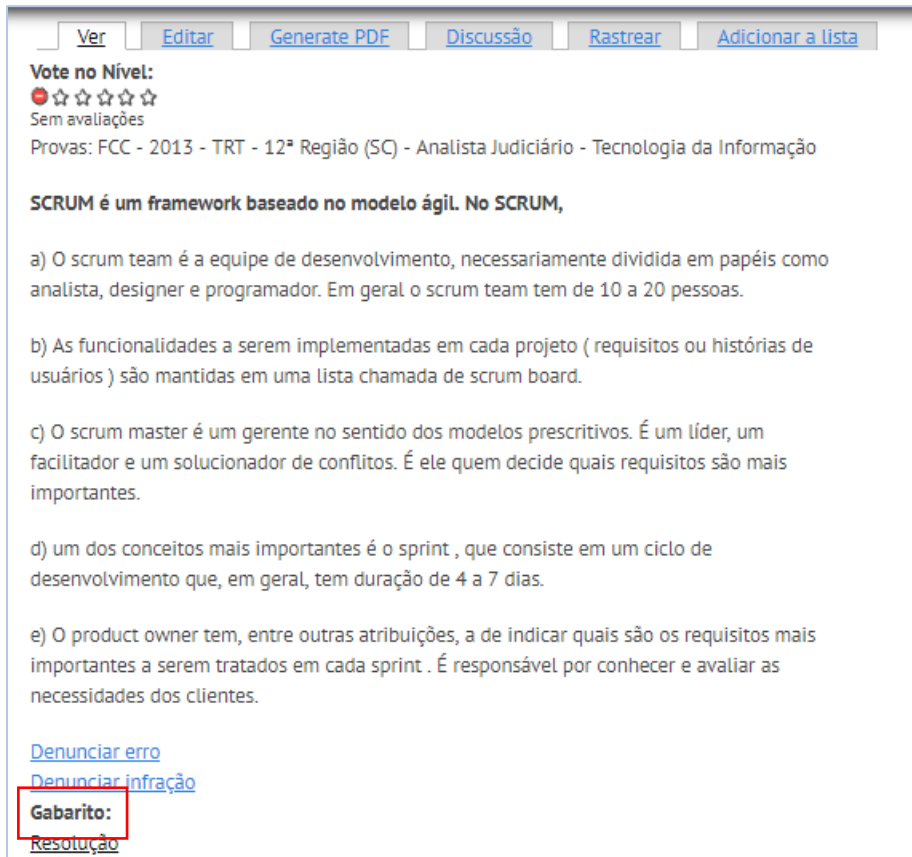
Como para esta história foram criados três critérios de aceitação, da mesma forma, para a sua validação, foram criados três testes de aceitação, conforme apresentados nos Quadros 54 a 56.

Quadro 54 - Testes de Aceitação para a história de usuário 3- Critério 1

Teste de Aceitação- Critério 1		
História:	3	
Número	Ação	Resultado Esperado
1	- O usuário está logado e clica em 'Adicionar Exercício' na página inicial.	- O sistema abre a página para que o usuário insira os dados do novo exercício: disciplina, termos, compartilhar, texto, resolução e gabarito.
2	- O usuário preenche os dados do novo exercício e clica em 'Salvar'.	- O sistema confirma a inserção do novo exercício e mostra em uma nova página a sua visualização.

Atendendo ao critério 1 da história 3, a Figura 37 apresenta um novo exercício inserido no sistema contendo, dentre outras informações, o campo 'Gabarito' (destaque em vermelho).

Figura 37 - Cadastro de um novo exercício com o campo 'Gabarito'



The screenshot shows a user interface for registering a question. At the top, there are buttons for 'Ver', 'Editar', 'Generate PDF', 'Discussão', 'Rastrear', and 'Adicionar a lista'. Below these is a 'Vote no Nível:' section with a red circle icon and five empty star icons, followed by the text 'Sem avaliações'. The question title is 'Provas: FCC - 2013 - TRT - 12ª Região (SC) - Analista Judiciário - Tecnologia da Informação'. The question text is 'SCRUM é um framework baseado no modelo ágil. No SCRUM,'. Below the text are five multiple-choice options (a) through (e) describing SCRUM roles and processes. At the bottom, there are links for 'Denunciar erro' and 'Denunciar infração'. The 'Gabarito:' field is highlighted with a red box, and the 'Resolução' field is visible below it.

Ver Editar Generate PDF Discussão Rastrear Adicionar a lista

Vote no Nível:
●☆☆☆☆
Sem avaliações

Provas: FCC - 2013 - TRT - 12ª Região (SC) - Analista Judiciário - Tecnologia da Informação

SCRUM é um framework baseado no modelo ágil. No SCRUM,

a) O scrum team é a equipe de desenvolvimento, necessariamente dividida em papéis como analista, designer e programador. Em geral o scrum team tem de 10 a 20 pessoas.

b) As funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuários) são mantidas em uma lista chamada de scrum board.

c) O scrum master é um gerente no sentido dos modelos prescritivos. É um líder, um facilitador e um solucionador de conflitos. É ele quem decide quais requisitos são mais importantes.

d) um dos conceitos mais importantes é o sprint , que consiste em um ciclo de desenvolvimento que, em geral, tem duração de 4 a 7 dias.

e) O product owner tem, entre outras atribuições, a de indicar quais são os requisitos mais importantes a serem tratados em cada sprint . É responsável por conhecer e avaliar as necessidades dos clientes.

[Denunciar erro](#)
[Denunciar infração](#)
Gabarito:
[Resolução](#)

O teste de aceitação criado para o segundo critério da história 3 pode ser visualizado no Quadro 55.

Quadro 55 - Testes de Aceitação para a história de usuário 3- Critério 2

Teste de Aceitação- Critério 2		
História:	3	
Número	Ação	Resultado Esperado
1	- O usuário está logado e clica em 'Meus Simulados' na página inicial.	- O sistema abre uma nova página com a opção para 'Adicionar Simulado'.
2	- O usuário preenche os dados e clica em Adicionar.	- O sistema abre a página do novo simulado e exibe um <i>link</i> para o usuário adicionar exercícios ao mesmo.
3	- O usuário clica no <i>link</i> .	- O sistema abre uma página para que o usuário insira a quantidade de exercícios que deseja para o simulado e o assunto.
4	- O usuário insere os dados e clica em 'Procurar'.	- O sistema abre uma página com a lista de exercícios do simulado gerado com as opções: 'Aceitar Simulado' e 'Gerar novamente'.
5	- O usuário clica na opção 'Gerar novamente'.	- O sistema abre uma nova janela para que o usuário insira a quantidade de exercícios que deseja para a lista do simulado e o assunto.
6	- O usuário insere os dados e clica em 'Confirmar'.	- O sistema abre uma página com a lista gerada com as opções : 'Aceitar Simulado' e 'Gerar novamente'.
7	- O usuário clica na opção 'Aceitar Simulado'.	- O sistema mostra o simulado criado.

Atendendo ao critério 2 da história 3, a Figura 38 apresenta um exercício do simulado criado para que o usuário possa 'Aceitar Simulado' ou 'Gerar Novamente' (destaque em vermelho).

Figura 38 - Exercício sugerido para um novo simulado criado

Provas: FCC - 2013 - TRT - 12ª Região (SC) - Analista Judiciário - Tecnologia da Informação

SCRUM é um framework baseado no modelo ágil. No SCRUM,

a) O scrum team é a equipe de desenvolvimento, necessariamente dividida em papéis como analista, designer e programador. Em geral o scrum team tem de 10 a 20 pessoas.

b) As funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuários) são mantidas em uma lista chamada de scrum board.

c) O scrum master é um gerente no sentido dos modelos prescritivos. É um líder, um facilitador e um solucionador de conflitos. É ele quem decide quais requisitos são mais importantes.

d) um dos conceitos mais importantes é o sprint , que consiste em um ciclo de desenvolvimento que, em geral, tem duração de 4 a 7 dias.

e) O product owner tem, entre outras atribuições, a de indicar quais são os requisitos mais importantes a serem tratados em cada sprint . É responsável por conhecer e avaliar as necessidades dos clientes.

[Leia mais](#) | 1 leitura

Gerar Novamente Aceitar Simulado

O teste de aceitação criado para o terceiro critério da história 3 pode ser visualizado no Quadro 56.

Quadro 56 - Testes de Aceitação para a história de usuário 3- Critério 3

Teste de Aceitação- Critério 3		
História:	3	
Número	Ação	Resultado Esperado
1	- O usuário está logado e clicou em 'Meus Simulados' na janela principal do site. - O usuário seleciona um simulado criado.	- O sistema abre uma página com as questões do simulado, mostrando para cada questão a opção 'Resposta'.
2	- O usuário clica no campo 'Resposta' de uma ou mais questões, selecionando a (s) alternativa (s) desejada (s). - O usuário clica no botão 'Corrigir Simulado'.	- O sistema mostra para cada questão o status da resposta (Correto ou Incorreto) e apresenta para o usuário o seu índice de acertos em porcentagens.

Atendendo ao critério 3 da história 3, a Figura 39 apresenta um exercício do simulado com o campo 'Resposta' (destaque em vermelho).

Figura 39 - Campo 'Resposta' de uma questão do Simulado

Resposta:

Provas: FCC - 2013 - TRT - 12ª Região (SC) - Analista Judiciário - Tecnologia da Informação

SCRUM é um framework baseado no modelo ágil. No SCRUM,

a) O scrum team é a equipe de desenvolvimento, necessariamente dividida em papéis como analista, designer e programador. Em geral o scrum team tem de 10 a 20 pessoas.

b) As funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuários) são mantidas em uma lista chamada de scrum board.

c) O scrum master é um gerente no sentido dos modelos prescritivos. É um líder, um facilitador e um solucionador de conflitos. É ele quem decide quais requisitos são mais importantes.

d) um dos conceitos mais importantes é o sprint , que consiste em um ciclo de desenvolvimento que, em geral, tem duração de 4 a 7 dias.

e) O product owner tem, entre outras atribuições, a de indicar quais são os requisitos mais importantes a serem tratados em cada sprint . É responsável por conhecer e avaliar as necessidades dos clientes.

[Denunciar erro](#)
[Denunciar infração](#)

1 leitura

A Figura 40 apresenta a tela com o resultado da resposta selecionada pelo usuário, bem como o resultado final do simulado feito pelo usuário, em porcentagem (destaque em vermelho).

Figura 40 - Resultado do simulado gerado

The screenshot displays a user interface for a simulation result. At the top left, a blue button labeled "Corrigir Simulado" is highlighted with a red box. Below it, the text "Acertos: 100%" and "Erros: 0%" is shown. A series of navigation links are present: "Remover da lista.", "Mover para cima.", "Mover para baixo.", "Mover para o topo.", "Mover para o fim.", and "Ver Exercício". Below these links, there is a "Vote no Nível:" section with a red circle icon and five empty star icons, with the text "Sem avaliações" underneath. A "Resposta:" section shows a dropdown menu with "E" selected and a green bar below it with the word "Correto", both highlighted with a red box. At the bottom, the text "Provas: FCC - 2013 - TRT - 12ª Região (SC) - Analista Judiciário - Tecnologia da Informação" is displayed, followed by the question "SCRUM é um framework baseado no modelo ágil. No SCRUM," and three multiple-choice options (a, b, c).

Corrigir Simulado

Acertos: 100%
Erros: 0%

[Remover da lista.](#) [Mover para cima.](#) [Mover para baixo.](#) [Mover para o topo.](#) [Mover para o fim.](#)
[Ver Exercício](#)

Vote no Nível:
☆☆☆☆☆
Sem avaliações

Resposta:
E ▾
Correto

Provas: FCC - 2013 - TRT - 12ª Região (SC) - Analista Judiciário - Tecnologia da Informação

SCRUM é um framework baseado no modelo ágil. No SCRUM,

a) O scrum team é a equipe de desenvolvimento, necessariamente dividida em papéis como analista, designer e programador. Em geral o scrum team tem de 10 a 20 pessoas.

b) As funcionalidades a serem implementadas em cada projeto (requisitos ou histórias de usuários) são mantidas em uma lista chamada de scrum board.

c) O scrum master é um gerente no sentido dos modelos prescritivos. É um líder, um