INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS NATAL-ZONA NORTE
CURSO TÉCNICO INTEGRADO EM INFORMÁTICA PARA INTERNET

GIOVANNA LORENA RODRIGUES DOS SANTOS

UM ESTUDO SOBRE PROCESSOS DE DESENVOLVIMENTO DE *SOFTWARE* PARA AMBIENTE ACADÊMICO

NATAL/RN
2017

GIOVANNA LORENA RODRIGUES DOS SANTOS

UM ESTUDO SOBRE PROCESSOS DE DESENVOLVIMENTO DE *SOFTWARE* PARA AMBIENTE ACADÊMICO

Relatório de Prática Profissional apresentado ao curso Técnico de Nível Médio Integrado em Informática para Internet do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática para Internet.

Orientador: Profº Edmilson Barbalho Campos Neto.

NATAL/RN
2017

GIOVANNA LORENA RODRIGUES DOS SANTOS

UM ESTUDO SOBRE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE PARA AMBIENTE ACADÊMICO

Relatório de Prática Profissional apresentado à Coordenação do Curso Técnico em Informática para Internet do *Campus* Natal-Zona Norte do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática para Internet.

Relatório de Prática Profissional apresentado e APROVADO em 08/12/2017, pela seguinte Banca Examinadora:

Edmilson Barbalho Campos Neto, Me.
Coordenador do Curso Técnico em Informática para Internet - IFRN

BANCA EXAMINADORA

Edmilson Barbalho Campos Neto, Me.
Orientador – IFRN

Alyana C. M. de Barros

Alyana Carindé Macêdo de Barros, Esp.
Examinadora Interna – IFRN

RESUMO

Os sistemas informatizados estão cada vez mais presentes no cotidiano da sociedade moderna, com demandas de soluções de softwares para os mais variados contextos e atividades-fim. Por essa razão, o desenvolvimento destes se tornou mais elaborado e planejado. Quando se pensa em desenvolver um *software* de alta qualidade, em um período determinado de tempo, dentro de um orçamento específico e por meio de uma equipe com integrantes que exercem diferentes funções, faz-se indispensável a utilização de um Processo de Desenvolvimento de *Software* (PDS). É de um adequado processo, portanto, que depende a qualidade do produto, o cumprimento de prazos e a satisfação do cliente. Não existe, entretanto, um único processo para todos os sistemas a serem desenvolvidos, uma vez que cada projeto exige demandas diferentes. Nesse contexto, o ambiente acadêmico forma profissionais mais preparados nessa área quando os discentes têm acesso a uma realidade prática de desenvolvimento de software com um PDS. Nesse sentido, o IFRN Campus Natal-Zona Norte possui um projeto de extensão denominado Fábrica de *Software* Escola (FaSEs) que trabalha, entre outras coisas, com a disciplina de Projeto de Desenvolvimento de Sistemas para Internet do curso de Informática para Internet, na qual os alunos devem desenvolver um projeto seguindo um PDS. Dessa forma, surge a necessidade de um Processo de Desenvolvimento de *Software* adequado à realidade acadêmica, considerando-se as suas limitações e diferenças em relação a indústria. Este trabalho propõe o FaSEs Process, um PDS baseado em metodologias ágeis para desenvolvimento de *software* e com uma documentação reduzida e adaptável, com objetivo de proporcionar uma experiência efetiva no aprendizado de Engenharia de *Software* nas turmas de 4º ano do IFRN Campus Natal-Zona Norte no contexto da FaSEs. Para a realização do PDS, foi feito um mapeamento sistemático com o intuito de ver na literatura o que já existe sobre PDS em ambiente acadêmico. Posteriormente, o processo foi definido e implementado na ferramenta *Eclipse Process Framework Composer* e disponibilizado na rede para que seja acessível a todos, portanto, também pode ser adaptado a outras realidades de outras instituições e projetos.

Palavras-chave: Processo de desenvolvimento de software. Ensino de engenharia de software. PDS acadêmico.

ABSTRACT

Computerized systems are increasingly present in the daily life of modern society, with demands for software solutions for the most varied contexts and end-activities. For this reason, the development of these has become more elaborate and planned. Thinking about developing high-quality software within a specific period, within a specific budget and through a team with members who perform different functions, it is essential to use a Software Development Process. It is on an appropriate process, therefore, that depends on product quality, meeting deadlines and customer satisfaction. There is, however, no single process for all systems to be developed, since each project demands different demands. In this context, the academic environment forms more educated professionals in this area when students have access to a practical reality of software development with a SDP. In this sense, the IFRN Campus Natal-Zona Norte has an extension project called the *Fábrica de Software Escola* (FaSEs), which works, among other things, with the discipline of *Projeto de Desenvolvimento de Sistemas para Internet* of *Informática para Internet* course, in which students should develop a project following a SDP. Thus, the need arises for a Software Development Process adequate to the academic reality, considering its limitations and differences in relation to the industry. This work proposes the FaSEs Process, a SDP based on agile methodologies for software development and with a reduced and adaptable documentation, aiming to provide an effective experience in the learning of Software Engineering in the 4th year classes of the IFRN Campus Natal-Zona Norte in the context of the FaSEs. For the realization of the SDP, a systematic mapping was done to see in the literature what already exists about SDP in an academic environment. Later, the process was defined and implemented in the Eclipse Process Framework Composer tool and made available on the network so that it can be accessed by all, so it can also be adapted to other realities of other institutions and projects.

Keywords: Software Development Process. Teaching Software Engineering. Academic PDS.

SUMÁRIO

1. INTRODUÇÃO

Cada vez mais, o desenvolvimento de software vem exigindo uma maior elaboração e planejamento de suas etapas, para que não haja perda de produtividade nas equipes em razão do aumento da demanda por soluções tecnológicas para os diversos setores da sociedade. Nesse contexto, surge a necessidade de um Processo de Desenvolvimento de Software (PDS), que define uma metodologia para atividades, ações e tarefas por meio das quais um produto de software é desenvolvido, visando otimizar os resultados da equipe (MALIK, 2008). O *OpenUp* e o *Scrum* são exemplos de metodologias de processo bem conhecidas e adotadas pela academia e indústria. Todavia, em função das especificidades do meio acadêmico -, tais como o nível de formação (técnica, graduação); a distribuição da carga horária da prática de PDS nos currículos dos cursos (semestral, anual); os recursos tecnológicos oferecidos aos discentes; e a natureza dos sistemas desenvolvidos (móveis, distribuídos, etc.), - é crucial a adaptação dos processos existentes para a utilização no ambiente acadêmico.

1.1. JUSTIFICATIVA

No contexto anteriormente apresentado, formar desenvolvedores com habilidade para trabalhar em equipes e adotar algumas das metodologias de desenvolvimento é um dos maiores desafios do ensino de engenheira de software (GIBBS, 1994). Ao encontro dessa problemática, Silva, Leite e Breitman (2004) investigaram que o ensino prático ainda é uma das formas mais eficientes de aprendizado. A exemplo desse método de ensino, Pontes et al. (2007) propõem o Processo Acadêmico Simplificado (PAS), que é baseado no *Rational Unified Process* (RUP) e no *eXtreme Programming* (XP), duas outras conhecidas metodologias, para utilização nas disciplinas de práticas de PDS em curso superior de análise de sistemas. Outro PDS acadêmico, o ENgAGED, é proposto por Battistella e Wangenheim (2016) para o desenvolvimento de jogos.

Diante disso, é ideal que a disciplina de Projeto de Desenvolvimento de Sistemas para Internet ofertada no curso Técnico Integrado em Informática para Internet, bem como a Fábrica de Software Escola (FaSEs) (CAMPOS NETO; LOPES; NASCIMENTO, 2017), tenham um processo de software adequado às suas demandas e necessidades, para melhor solucionar os problemas reais da comunidade acadêmica e da indústria.

1.2. OBJETIVO GERAL

Este trabalho tem como objetivo geral a elaboração de um processo de desenvolvimento de software, baseado em metodologias difundidas na área, que se adeque à realidade da disciplina de Projeto de Desenvolvimento de Sistemas para Internet no curso Técnico Integrado em Informática para Internet do Campus Natal-Zona Norte, suprindo a necessidade da Divisão Acadêmica da Fábrica de Software Escola (FaSEs) (CAMPOS NETO; LOPES; NASCIMENTO, 2017). Nesse sentido, os alunos dessa instituição poderão ter uma vivência que lhes proporcione as condições necessárias para se tornarem técnicos aptos a trabalhar no desenvolvimento de softwares.

1.3. METODOLOGIA

Para elaboração do processo proposto, foram realizadas as etapas descritas a seguir:

- Realização de um mapeamento sistemático: esta etapa teve o objetivo de investigar o que já existe na literatura acerca de processos de desenvolvimento de softwares em ambiente acadêmico. Este mapeamento buscou, em especial, identificar sobre cada PDS acadêmico encontrado: (i) a metodologia que se baseia (ex. RUP, Scrum, etc.); (ii) a natureza de utilização (tipos de sistemas desenvolvidos, tais como desenvolvimento

móveis, de jogos, sistemas pedagógicos, etc.); (iii) a nível de formação do curso (técnico, tecnológico, etc.); e (iv) detalhes da estrutura do processo (fases, tarefas e artefatos). Para tanto, foi realizada uma busca manual nos anais dos últimos 10 anos (2006-2016) dos cinco mais relevantes veículos de publicação da área de ensino de computação no Brasil: Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola (WIE); *Workshop* sobre Educação em Informática (WEI); Revista Novas Tecnologias da Informação (RENOTE) e na Revista Brasileira de Informática na Educação (RBIE). Nessa etapa, também foi feita uma avaliação comparativa e qualitativa dos processos encontrados;

- Proposição de um processo de desenvolvimento de *software* específico para o ambiente acadêmico do curso técnico em informática para internet do campus Natal-Zona Norte: nessa etapa, foram definidas as fases, atividades, tarefas e artefatos do processo, tomando como referência os processos de metodologia ágil existentes, bem como a experiência dos alunos do 4º ano de Informática para Internet do corrente ano;
- Representação do processo com *Eclipse Process Framework Composer*: nessa etapa, o processo elaborado foi implementado na ferramenta EPF Composer para que pudesse se tornar público e facilmente acessível.

1.4. PRINCIPAIS RESULTADOS

Nessa subseção serão apresentados os principais resultados do desenvolvimento do trabalho, de acordo com o que foi previsto na Metodologia.

1.4.1. **Realização do Mapeamento Sistemático**

- Após busca manual nos anais dos seguintes eventos e revistas: SBIE, WEI, WIE, HOLOS, RENOTE e RBIE, foram encontrados oito artigos que propõem um PDS;
- As metodologias de desenvolvimento mais utilizadas pelos processos propostos foram: *Rational Unified Process* (RUP) e *eXtreme Proc*ess (XP);
- A maioria dos processos possui de 4 a 5 fases;
- Apenas dois dos processos são voltados especificamente para produção em ambiente acadêmico.

1.4.2. **Proposição do FaSEs Process**

- Processo dividido em 4 fases: iniciação, elaboração, construção e transição;
- Definição das atividades, tarefas, artefatos e papéis de cada fase.

1.4.3. **Representação do Processo com EPF Composer**

- Implantação das fases, atividades, tarefas, artefatos e papéis do processo no EPF Composer;
- Criação de guias para os artefatos;
- Montagem dos diagramas das fases, atividades e tarefas;
- Publicação do processo.

1.5. ORGANIZAÇÃO DO DOCUMENTO

Este documento está organizado, a partir dessa seção, em 4 seções. Na segunda seção será discutida a fundamentação teórica do trabalho; a terceira seção

apresentará os resultados do mapeamento sistemático realizado; a quarta seção discorrerá sobre os resultados da criação do FaSEs Process e a quinta apresentará as considerações finais do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Nessa seção do trabalho serão apresentados conceitos e definições necessários para realização e compreensão desse trabalho. Divide-se em 5 subseções, que tratam dos conceitos de: PDS, Metodologias Ágeis, Quadro de Modelo de Negócios, *Eclipse Process Framework* e Fábrica de Software Escola (FaSEs), nessa ordem.

2.1. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE (PDS)

Os sistemas informatizados estão cada vez mais presentes no cotidiano da sociedade moderna, com demandas de soluções de softwares para os mais variados contextos e atividades-fim (SOMMERVILLE, 2011). Por essa razão, o desenvolvimento destes se tornou, ao decorrer dos anos, mais elaborado e planejado. Quando se pensa em desenvolver um software de alta qualidade, em um período determinado de tempo, dentro de um orçamento específico e por meio de uma equipe com integrantes que exercem diferentes funções, faz-se indispensável a utilização de um PDS (PMBOK, 2004). A definição de processo de software está diretamente relacionada a uma metodologia para atividades, ações e tarefas por meio das quais um produto de software é desenvolvido (PRESSMAN, 2011).

Um PDS especifica as atividades que devem ser realizadas durante o desenvolvimento do software, bem como as tarefas a serem cumpridas e por quem elas serão cumpridas, além dos artefatos que serão gerados com o cumprimento dessas tarefas. Essas atividades são agrupadas em fases, cada fase tem um objetivo a ser alcançado no ciclo de vida do processo (SCOTT, 2003).

É de um adequado processo, portanto, que depende a qualidade do produto, o cumprimento de prazos e a satisfação do cliente. Não existe, entretanto, um único processo de software para todos os sistemas a serem desenvolvidos, tendo em vista que cada projeto exige demandas diferentes de tarefas e entregas (AMBLER, 2004).

2.2. METODOLOGIAS ÁGEIS

Com o passar dos anos, a competitividade na indústria de software se tornou cada vez mais alta, por isso, as empresas de softwares passaram buscar sempre inovação e velocidade em oferecer soluções, mas para que essas soluções sejam de qualidade, é necessário um processo adequado (PRIKLADNICKI; WILLI; MILANI, 2014). Nesse contexto, a partir dos anos 1990, começaram a surgir os Métodos Ágeis, o que foi, de certa forma, uma quebra de paradigmas. De acordo com Prikladnicki, Willi e Milani (2014), a diferença dos métodos ágeis para os métodos tradicionais consiste no "enfoque maior nas pessoas e não em processo, e o seu conjunto de valores, princípios e práticas". Dessa forma, abre-se espaço para corresponder a necessidade de respostas rápidas às mudanças no mercado.

Em 2001 foi estabelecido o Manifesto Ágil[1], que declara a valorização de: (i) indivíduos e interações mais que processos e ferramentas; (ii) software funcionando mais do que documentação abrangente; (iii) colaboração com o cliente mais que negociação de contratos; e (iv) responder a mudanças mais que seguir um plano. Pode-se citar como exemplos de processos que adotam a metodologia ágil o *Scrum* e o *OpenUP*.

2.3. QUADRO DE MODELO DE NEGÓCIOS – CANVAS

Conforme Osterwalder e Pigneur (2013), "um Modelo de Negócios define a lógica de criação, entrega e captura de valor por parte de uma organização". O Quadro de Modelo de Negócios ou *Business Model Canvas*, foi inicialmente proposto por Alexander Osterwälder e consiste em uma ferramenta de gerenciamento estratégico para esboçar e descrever modelos de negócios (MOURA, 2014).

[1] Disponível em: <www.manifestoagil.com.br>

Osterwalder e Pigneur (2013) definem 9 componentes do quadro de modelo de negócios: (i) segmentos de clientes, (ii) proposta de valor, (iii) canais, (iv) relacionamento com clientes, (v) fontes de receita, (vi) recursos principais, (vii) atividades-chave, (viii) parcerias principais e (ix) estrutura de custos. Outras propostas de componentes foram feitas e adequadas a diferentes realidades, a que foi adotada para utilização no PDS proposto pode ser visualizada na Figura 1.

Figura 1 – Modelo Canvas adotado pelo FaSEs Process

Fonte: Elaborado pela autora (2017).

2.4. ECLIPSE PROCESS FRAMEWORK – EPF COMPOSER

O *Eclipse Process Framework Composer* (EPF Composer)[2] é uma ferramenta para descrição e especificação de processos de desenvolvimento de *software*. Esse *software* permite a especificação e a manipulação dos componentes de um processo, como excluir, adicionar e modificar atividades, tarefas, fluxos, papéis, guias etc. O maior exemplo de processo que está especificado em EPF Composer é o *OpenUP.*

[2] Disponível em: <www.eclipse.org/epf>

De acordo com EPF (2017), existem dois principais objetivos do *Eclipse Process Framework*: (i) fornecer estrutura e ferramentas exemplares para engenharia de processos de software, visando a criação de métodos e processos, gerenciamento de bibliotecas, configuração e publicação de um processo; (ii) proporcionar conteúdo de processo exemplar para uma variedade de processos de desenvolvimento e gerenciamento de software que suportam desenvolvimento iterativo, ágil e incremental e aplicável a um amplo conjunto de plataformas e aplicações de desenvolvimento.

2.5. FÁBRICA DE SOFTWARE ESCOLA (FASES)

Fábrica de *Software* pode ser definida como:

> Um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial, orientado para o atendimento a múltiplas demandas de natureza e escopo distintas, visando à geração de produtos de software, conforme os requerimentos documentados dos usuários e/ou clientes, da forma mais produtiva e econômica possível. (FERNANDES; TEIXEIRA, 2004).

No contexto acadêmico, a Fábrica de *Software* pode ser uma forma inovadora de aliar teoria e prática visando uma melhor experiência de aprendizado para os alunos. Nessa perspectiva, a FaSEs, acrônimo para Fábrica *Software* Escola, é um projeto de extensão do IFRN Campus Natal-Zona Norte desenvolvido com o intuito de possibilitar aos alunos a aplicação dos conhecimentos de Engenharia de *Software* trabalhados em sala de aula, para que possam criar soluções de *software* para demandas da comunidade acadêmica ou industrial (CAMPOS NETO; LOPES; NASCIMENTO, 2017). Nesse sentido, o projeto busca promover uma vivência de um ambiente mais próximo da realidade de produção de *software* aos alunos dos cursos de tecnologia. O resumo dos resultados do ano letivo de 2016 está ilustrado na Figura 2.

Figura 2 – Visão geral dos resultados da FaSEs em 2016

Fonte: (CAMPOS NETO; LOPES; NASCIMENTO, 2017).

A FaSEs, como pode ser visto na Figura 2, possui as divisões acadêmica e de inovação. A Divisão de Inovação corresponde aos projetos de pesquisa executados na área de informática, enquanto a Divisão Acadêmica corresponde aos projetos desenvolvidos em sala de aula, na disciplina de Projeto de Desenvolvimento de Software. O PDS proposto nesse trabalho é para o contexto da Divisão Acadêmica.

3. MAPEAMENTO SISTEMÁTICO SOBRE PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE EM AMBIENTE ACADÊMICO

Nessa seção serão apresentados os resultados da realização do mapeamento sistemático sobre PDS em ambiente acadêmico.

3.1. INTRODUÇÃO

Alguns trabalhos já mapearam técnicas de ensino de engenharia de *software*, tal como em Santos et al. (2014), mas não focaram em identificar processos de desenvolvimento de software para o ambiente acadêmico. Nesse sentido, esse trabalho realizou um mapeamento sistemático baseado nas diretrizes de KITCHENHAM et al. (2014), com o propósito de identificar as metodologias de processos de desenvolvimento de software que estão sendo propostas e/ou utilizadas em ambientes acadêmicos.

3.2. PROTOCOLO

3.2.1. **Questões de pesquisa**

O objetivo desse trabalho é que os resultados da pesquisa possam auxiliar no aprimoramento das práticas de PDS adotadas por instituições de ensino e/ou professores que atuam no ensino de engenharia de software nessas instituições, além de corroborar na criação de um PDS adequado a realidade acadêmica da disciplina de Projeto de Desenvolvimento de Sistemas para Internet cursada pelos alunos do 4º

ano de Informática para Internet no IFRN-ZN. Para isso, foram elaboradas as seguintes questões de pesquisa no intuito de analisar os trabalhos selecionados:

- QP1. Em que metodologias o processo se baseia?;
- QP2. Qual a natureza de utilização do processo?;
- QP3. Qual o nível de formação do curso em que o processo foi desenvolvido?;
- QP4. Quais os detalhes da estrutura do processo?

A QP1 tem a intenção de identificar as metodologias utilizadas para a construção do processo (ex. RUP, *Scrum*, etc.), e constatar se há alguma predominância. A QP2 serve para identificar se o processo é utilizado para desenvolvimento de sistemas móveis, de jogos, sistemas de cunho pedagógicos, etc. A QP3, por sua vez, diferenciará o nível de formação no qual o processo está inserido (técnica, graduação, tecnológica). Por fim, a QP4 serve para definir a estrutura dos processos propostos, identificando fases, etapas, artefatos etc.

3.2.2. **Critérios de inclusão e exclusão**

Os critérios de inclusão e exclusão foram definidos de acordo com a capacidade do trabalho em responder às questões de pesquisa, afim de que os artigos selecionados sejam relevantes para alcançar o objetivo da pesquisa. O quadro 1 descreve os critérios estabelecidos.

Quadro 1 – Critérios de inclusão e exclusão

Critérios de inclusão	Critérios de exclusão
I1. Artigo com pelo menos 4 páginas I2. Artigo publicado a partir de 2005 I3. Artigo que propõe um novo PDS I4. Processo que foi utilizado/testado	E1. Artigo que não descreve a estrutura do processo proposto

Fonte: Elaborado pela autora (2017).

3.2.3. **Coleta de dados**

Os artigos a serem analisados foram selecionados por busca manual com a chave de busca “processo de desenvolvimento de software AND ensino”, nos anais de 2005 a 2016 dos seguintes veículos de publicação da área de ensino de computação no Brasil: Simpósio Brasileiro de Informática na Educação (SBIE); Workshop de Informática na Escola (WIE); Workshop sobre Educação em Informática (WEI); Revista Novas Tecnologias da Informação (RENOTE); e Revista Brasileira de Informática na Educação (RBIE).

3.2.4. **Dados extraídos**

A fim de analisar os trabalhos selecionados conforme o protocolo descrito anteriormente, foram extraídos dados gerais e dados que respondem às questões de pesquisa propostas. No quadro 2, estão descritos os dados extraídos e suas relações com as questões de pesquisa.

Quadro 2 – Dados extraídos dos trabalhos selecionados

Dado	Questão de pesquisa
Autor	-
Ano	-
Evento	-
Título	-
Instituição de produção	-
Metodologia utilizada	QP1
Natureza de utilização	QP2
Nível de formação	QP3
Número de fases do processo	QP4
Papéis do processo	QP4
Atividades do processo	QP4

Fonte: Elaborado pela autora (2017).

3.2.5. **Resultados**

Após realização da busca manual conforme os métodos já descritos, foram selecionados oito artigos a serem discutidos neste trabalho. As informações gerais acerca desses artigos estão descritas no quadro 3.

Quadro 3 – Descrição dos trabalhos selecionados

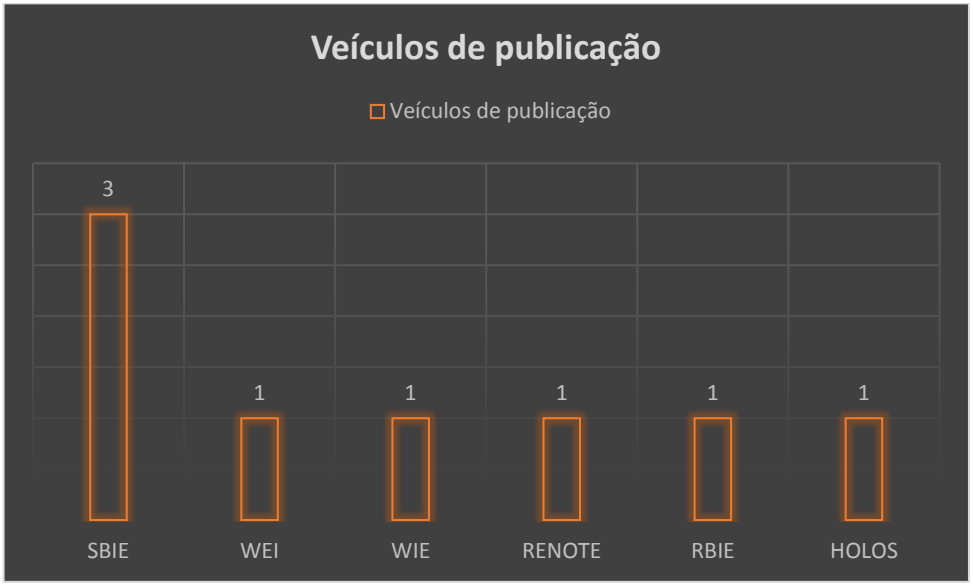
ID	Título	Evento	Ano
P1	Modelo Inclusivo de Desenvolvimento e Avaliação de Objetos de Aprendizagem Acessíveis voltados para o Ensino Superior em Computação	SBIE	2016
P2	PROCESSO ACADÊMICO SIMPLIFICADO: UMA PROPOSTA DE PROCESSO PARA O CEFET-RN/DATINF	HOLOS	2006
P3	Processo de Desenvolvimento do jogo sério Missão Aedes: relações entre objetivos pedagógicos, ludicidade e implicações de design	SBIE	2016
P4	Processo de Desenvolvimento de Software Educacional: proposta e experimentação	RENOTE	2005
P5	Aprendizagem Baseado em Metodologias Ágeis e Scaffoldings	RBIE	2010
P6	PDS-E: Em direção a um processo para desenvolvimento de Software Educacional	WIE	2005
P7	SimpleWayProcess: Da Academia à Indústria de Software	WEI	2012
P8	ENgAGED: Um Processo de Desenvolvimento de Jogos	SBIE	2016

Fonte: Elaborado pela autora (2017).

3.2.6. **Visão geral dos resultados**

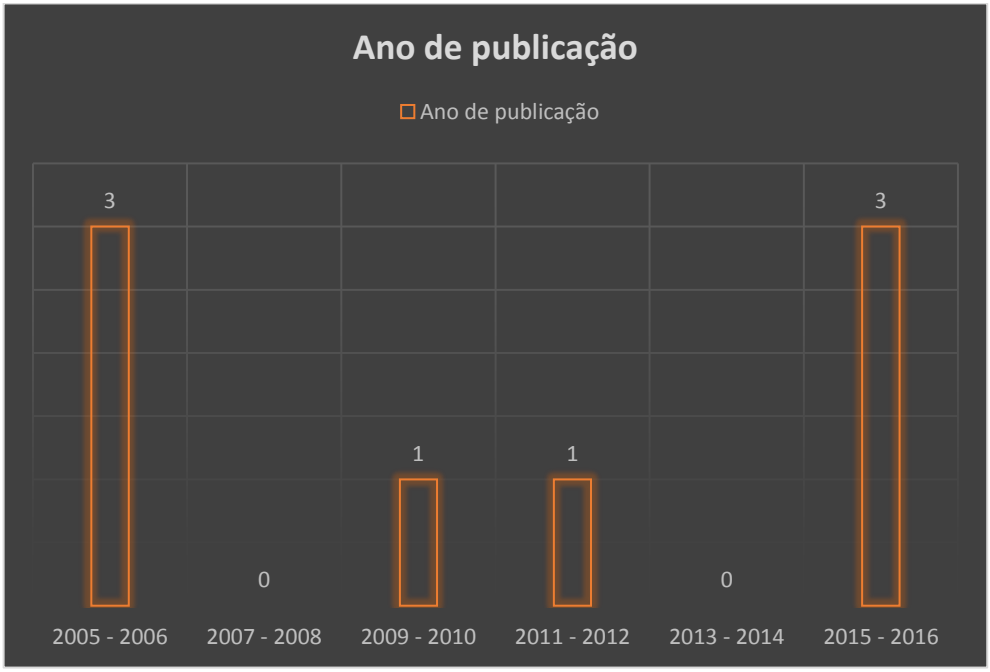
Os Gráficos 1, 2 e 3 mostram a quantidade de trabalhos por veículos de publicação, ano de publicação e região de produção, respectivamente. Em se tratando da primeira classificação, verifica-se que o veículo no qual mais se publicou sobre o assunto foi o SBIE, com o total de 3 trabalhos dos 8 selecionados, enquanto cada um dos outros veículos só possui 1 trabalho. Quanto ao ano de publicação, nos períodos de 2005 a 2006 e 2015 a 2016 foram publicados 6 dos 8 selecionados. Por fim, as regiões onde mais foram produzidos os trabalhos são a Nordeste e a Sul, sendo que apenas a região Centro-Oeste não produziu nenhum.

Gráfico 1 – Número de trabalhos por veículo de publicação



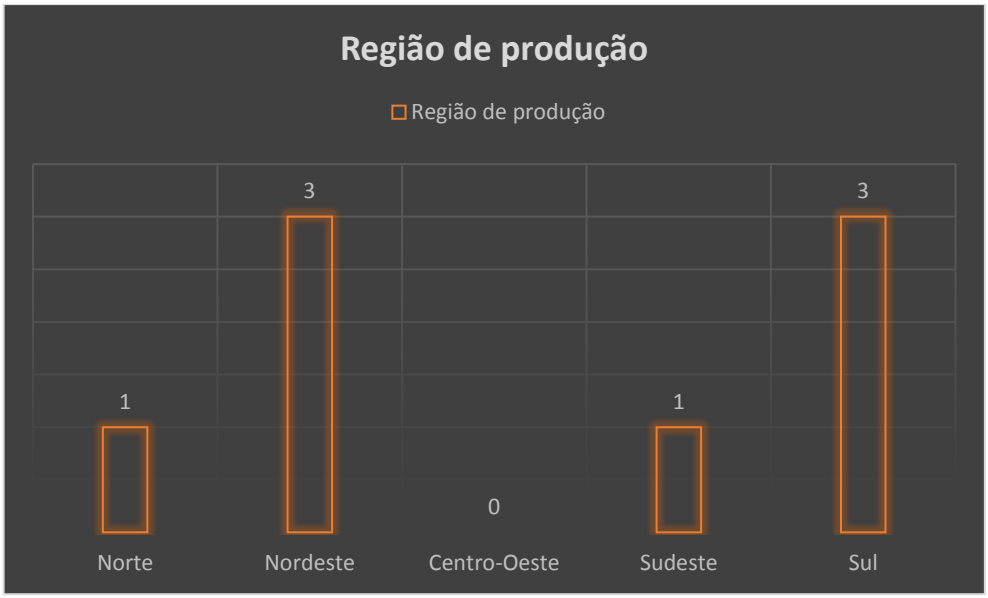
Fonte: Elaborado pela autora (2017).

Gráfico 2 – Número de trabalhos por ano de publicação



Fonte: Elaborado pela autora (2017).

Gráfico 3 - Número de trabalhos por região de produção



Fonte: Elaborado pela autora (2017).

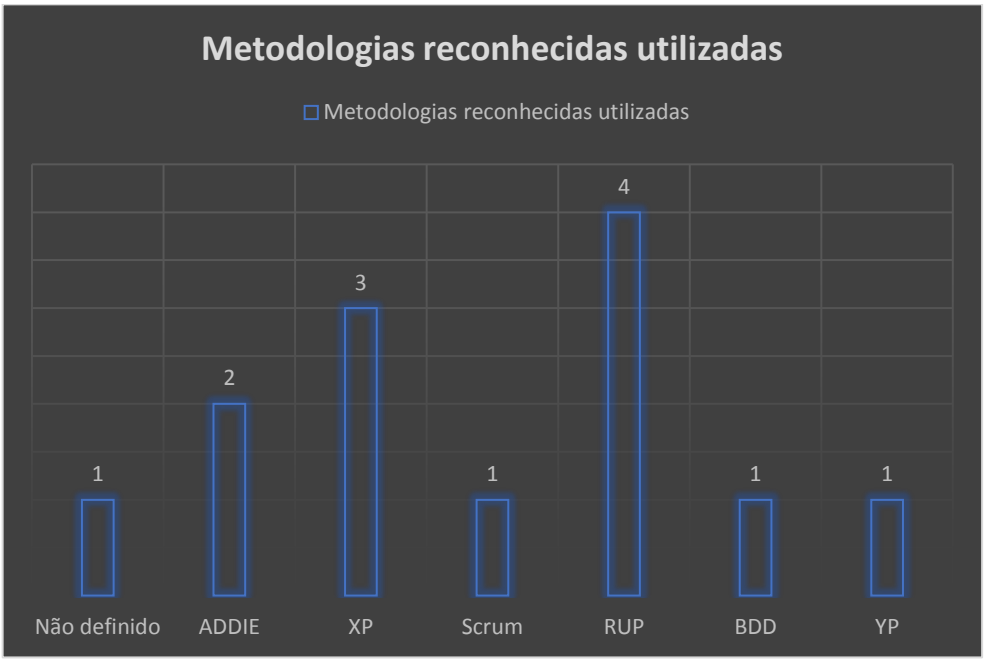
3.2.7. **Metodologia base dos processos (QP1)**

Para elaborarem os processos que sugeriram, os trabalhos tiveram como base algumas metodologias de PDS já reconhecidas e utilizadas na indústria. Entre elas, foram escolhidas 6 para representação gráfica (Gráfico 4), tendo em vista que se constituem as mais reconhecidas. Apenas um dos processos não especificou qual metodologia utilizou, enquanto a maioria utilizou mais de um deles. Os métodos mais utilizados foram o *Rational Unified Process* (RUP) e o *eXtreme Process* (XP). Outras metodologias foram utilizadas: *Scrum, Behaviour-Driven Development* (BDD), *easYProcess* (YP) e *Analysis, Design, Development, Implementation and Evaluation* (ADDIE). Este último não é um PDS, mas um modelo de Design Instrucional, utilizado pelos trabalhos que propuseram processos para o desenvolvimento de jogos educacionais.

Os processos propostos pelos trabalhos P2 e P7, que mais se aproximam da nossa proposta, utilizaram RUP, XP e *Scrum* e trabalham com o modelo iterativo-incremental. O RUP é um processo de caráter complexo (RUP, 2017), por isso os processos propostos são inspirados nele, mas com adaptações que o simplificam, aproveitando os elementos que se adequam às realidades específicas. O XP, por sua

vez, é mais leve e flexível (XP, 2017), ideal para trabalhar com requisitos dinâmicos. A metodologia *Scrum* é um método ágil muito difundido na indústria (SCRUM, 2017).

Gráfico 4 - Número de trabalhos por metodologias utilizadas

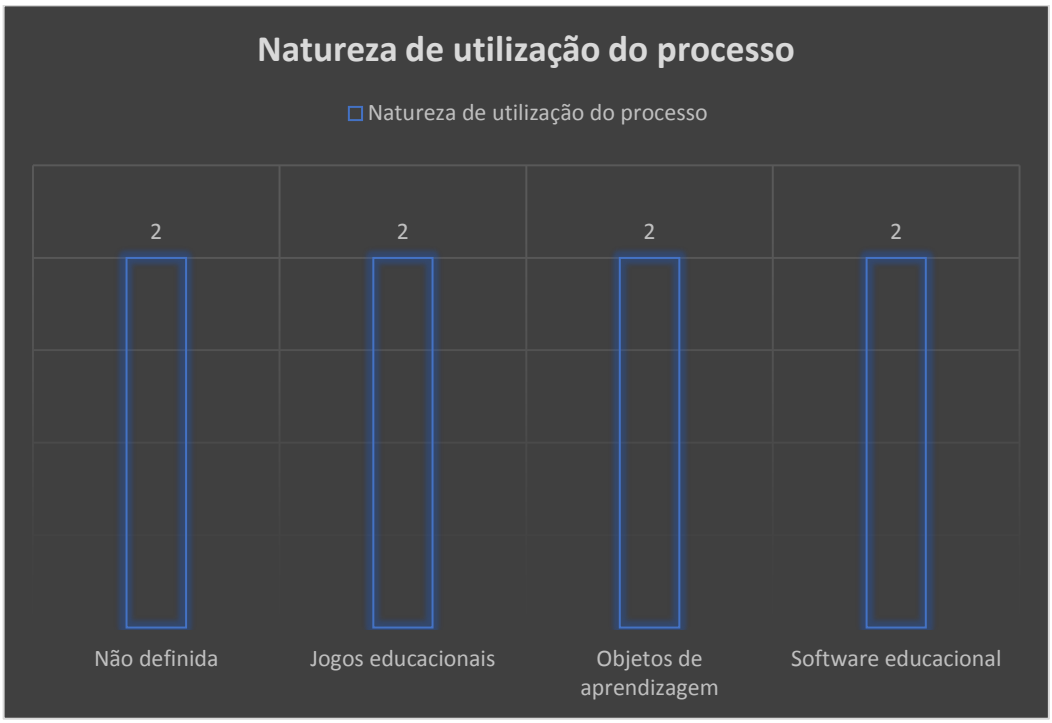


Fonte: Elaborado pela autora (2017).

3.2.8. **Natureza de utilização dos processos (QP2)**

A maioria dos processos propostos nos artigos selecionados tem cunho educativo como natureza de utilização, apenas com algumas diferenças de termos e conceitos. No Gráfico 5, pode-se observar as quatro classificações feitas: (i) não definida, (ii) jogos educacionais, (iii) objetos de aprendizagem e (iv) software educacional, sendo que cada uma corresponde a 25% dos trabalhos. Dessa forma, observa-se que 75% correspondem à natureza de cunho educacional e apenas 25% não define a natureza de utilização. Estes dois, por sua vez, possuem semelhanças que serão discutidas na próxima sessão.

Gráfico 5 - Número de trabalhos por natureza de utilização do processo

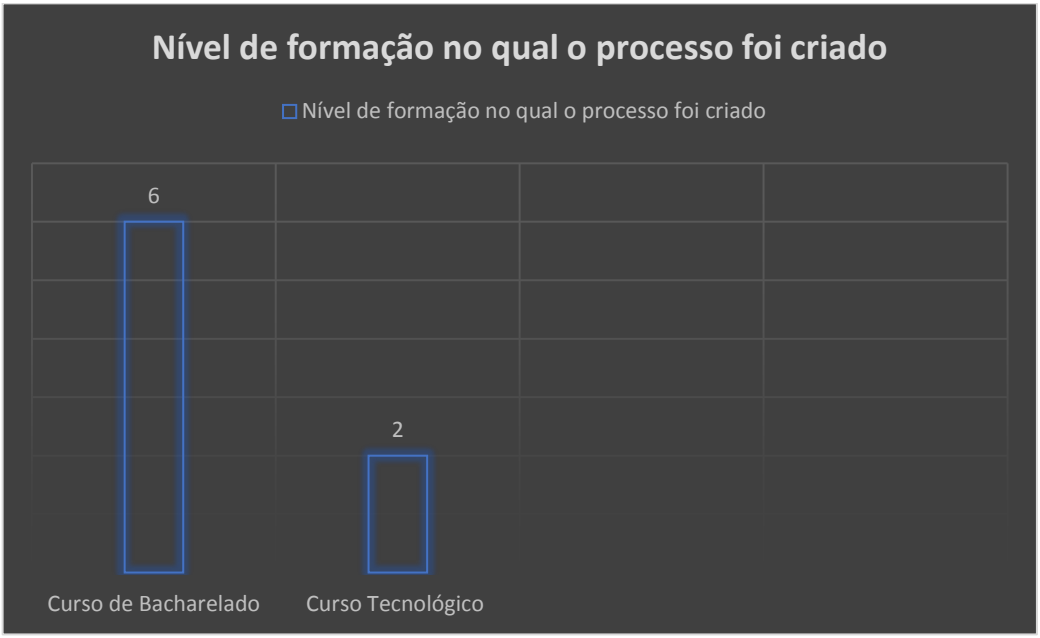


Fonte: Elaborado pela autora (2017).

3.2.9. **Nível de formação do curso em que o processo foi desenvolvido (QP3)**

A maioria dos processos foram produzidos em ambiente de cursos de bacharelado, como pode ser visualizado no gráfico 6. Apenas 2 dos processos foram produzidos e utilizados no curso de Tecnologia em Análise e Desenvolvimento de Sistemas, o qual forma tecnólogos. Esses dois últimos são, na verdade, os que mais se aproximam à proposta deste trabalho. Outros 2 processos foram produzidos por alunos da graduação, mas com o intuito de serem utilizados por uma comunidade externa e não em ambiente acadêmico. Ademais, apenas 1 especificou que o processo seria utilizado dentro de uma disciplina do curso. Entretanto, nenhum dos processos foi produzido e/ou utilizado em nível técnico de ensino.

Gráfico 6 - Número de trabalhos por nível de formação

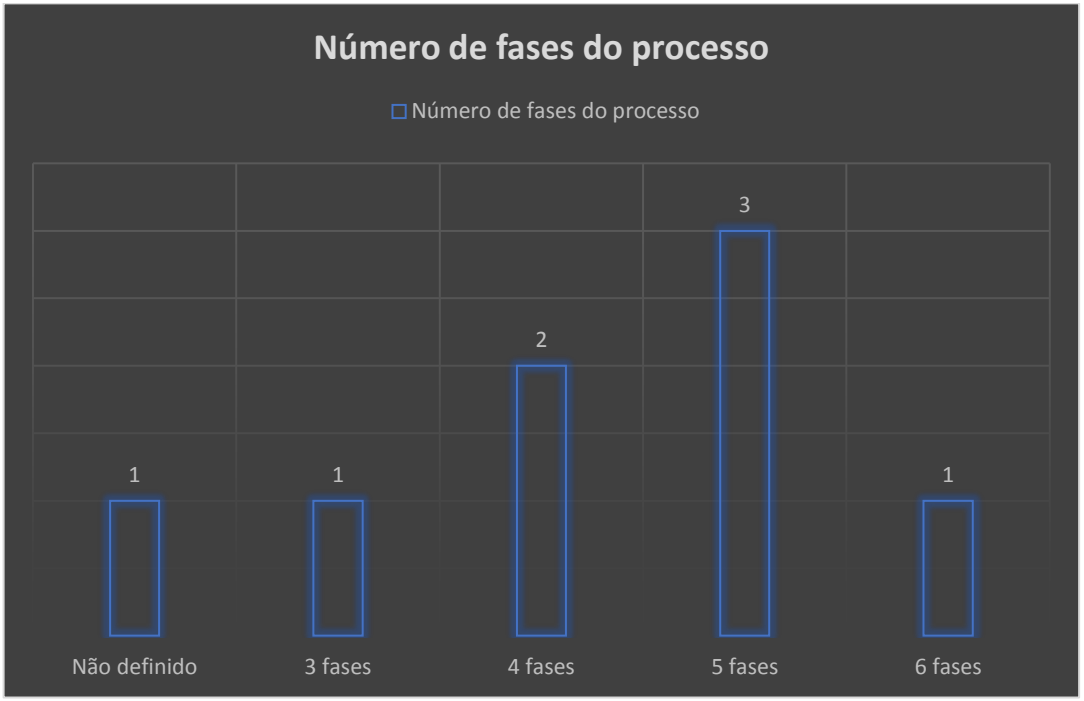


Fonte: Elaborado pela autora (2017).

3.2.10. **Detalhes da estrutura dos processos (QP4)**

Conforme apresentado no gráfico 7, os processos estudados propõem predominantemente de 4 a 5 fases. Apenas 2 sugerem 3 e 6 fases, sendo que um outro não define quantidade fases. Genericamente, as primeiras fases são de definição do projeto, levantamento de requisitos e definição de papéis, as intermediárias, são de construção do produto e, as últimas, fases de teste e implementação. Apesar dessa visão geral ser perceptível, cada processo tem suas diferenças de acordo com as demandas as quais se propõe atender.

Gráfico 7 - Número de trabalhos por número de fases



Fonte: Elaborado pela autora (2017).

Em se tratando de atores e atividades, poucos processos são específicos. Por exemplo, o processo proposto pelo trabalho P4 foi bem específico em relação a toda estrutura do processo, pois define 4 fases: concepção, elaboração/construção, finalização e viabilização; 4 atores: profissionais da educação, profissionais da área computacional, designer e aluno; e várias atividades para cada fase, como especificar incremento, construir protótipo, avaliar protótipo etc.

3.2.11. **Discussão dos resultados**

Nessa sessão serão apresentadas limitações observadas diante dos dados levantados com os trabalhos estudados, como também indicações ao que poderá ser contemplado no processo proposto por este trabalho.

- Natureza de utilização de cunho educacional – 6 dos 8 processos estudados são voltados para a construção de softwares de ensino ou jogos

educativos. Esse ponto, portanto, inviabiliza a utilização desses processos no contexto em questão, pois limitaria os alunos a produzirem um software dessa natureza, além dos próprios processos serem complexos e obedecerem a questões específicas para construção de softwares educativos. Apenas os trabalhos P2 e P7 não especificaram a natureza de utilização, não deixando claro em que casos os processos propostos são mais indicados para uso. Fazer essa especificação é essencial, uma vez que, como dito anteriormente, tipos de softwares diferentes exigem, muitas vezes, demandas muito diferentes a serem contempladas pelo processo utilizado.

- Nenhuma proposta para a modalidade de Ensino Médio Integrado – Todos os processos propostos nos trabalhos levantados foram feitos por alunos de cursos de nível superior. Os trabalhos P2 e P7 até foram de estudantes do Instituto Federal (IFRN e IFPB, respectivamente), mas do curso Tecnologia em Análise e Desenvolvimentos de Sistemas, de nível Superior. Na verdade, os outros 6 processos não foram feitos para serem utilizados regularmente no cumprimento de uma disciplina, mas foram propostas feitas por alunos para a construção de apenas um tipo de software específico (softwares de caráter educacional). O presente trabalho propõe a elaboração de um processo que seja utilizado pelos alunos, sendo adequado à realidade dos estudantes de ensino médio do IFRN (mais especificamente dos estudantes do *campus* Natal-Zona Norte). Essa realidade difere da dos cursos de nível Superior, tendo em vista que a carga horária é limitada, pois é dividida entre as disciplinas do núcleo comum e as de formação profissional. Além disso, o conteúdo referente a PDS só é estudado em uma disciplina no último ano do curso.
- Ambiente para acesso ao processo – Apenas os trabalhos P2 e P7 deixam explícito que disponibilizam ambientes de acesso ao processo, nos quais estão descritos todas as fases, tarefas e os seus respectivos atores. Esse é um passo importante para o processo que será proposto, pois ele deve ficar disponível a toda comunidade e principalmente aos alunos e professores.
- Ausência de indicações de ferramentas – Os trabalhos não definem que ferramentas podem/devem ser utilizadas nas diversas fases e atividades

do processo que propõem. Além disso, algumas das atividades são burocráticas e podem tornar o trabalho muito cansativo e até inviável para o contexto em questão neste trabalho, visto que a carga-horária da disciplina PDSI é de 2 horas-aula semanais. Para solucionar esse problema, o processo a ser elaborado definirá as entradas e saídas de cada fase com plataformas alternativas de fácil utilização.

3.2.12. **Conclusões do mapeamento sistemático**

Diante dos dados extraídos apresentados e das discussões propostas, entende-se que os trabalhos encontrados na literatura são, em sua maioria, voltados à produção de softwares de natureza educacional e, mesmo os que não o são, por serem propostos para ambientes de Nível Superior, não contemplam a necessidade dos alunos do Ensino Médio Integrado do IFRN. Dessa forma, faz-se ideal a elaboração de um processo que possa ser disponibilizado para o uso por professores nas futuras turmas de 4º do curso Informática para Internet e assim colaborar para a formação de profissionais aptos a desenvolver sistemas na indústria.

4. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE PARA SOFTWARE ESCOLA (FASES)

Nessa seção serão apresentados os procedimentos e resultados da proposição do FaSEs Process: a sua estruturação, bem como suas atividades, tarefas e seus artefatos gerados. A primeira subseção corresponde a estrutura geral do processo e, posteriormente, cada subseção corresponde à descrição da estrutura de uma das quatro fases do processo: iniciação, elaboração, construção e transição, respectivamente.

4.1. ESTRUTURA GERAL DO FASES PROCESS

Basicamente, o processo proposto tem 4 divisões: fases, atividades, tarefas e artefatos. Cada fase é dividida em atividades, as quais são divididas em tarefas, que, por sua vez, geram artefatos. Também foram definidos guias para os artefatos gerados nas tarefas. Esses guias podem ser exemplos ou *templates* e alguns possuem indicações de ferramentas e de tutoriais.

Com base nos resultados do Mapeamento Sistemático realizado e uma vez que o ano letivo no IFRN é divido em 4 bimestres, o processo proposto foi dividido em 4 fases: Iniciação, na qual busca-se responder à pergunta “O que será feito?”; Elaboração, na qual a pergunta é “Como será feito?”; Construção, que corresponde ao desenvolvimento real do projeto descrito e elaborado anteriormente; e Transição, quando ocorre os últimos ajustes e a implantação.

Também foram decididos quais os papéis presentes no processo: analista, arquiteto, desenvolvedor, *designer*, gerente, *stakeholder* e *tester*. Cada um desses papéis desempenha várias tarefas, mas no grupo em sala de aula, o mesmo aluno varia de papel a papel de acordo com a mudança de tarefa, atividade ou fase.

Ademais, tendo em vista a carga horária de apenas 2 horas-aula por semana na disciplina de Projeto de Desenvolvimento de Sistemas para Internet, o processo

proposto visa reduzir a documentação gerada, mas manter uma documentação que seja suficiente para o controle da sala de aula e que também possa ser adaptável de acordo com a turma e o professor. No quadro 4, pode-se observar alguns dos artefatos do processo, bem como em que documento ele foi inspirado, a metodologia de origem e a justificativa para a adaptação ou manutenção do artefato.

Quadro 4 – Inspiração de alguns artefatos do FaSEs Process

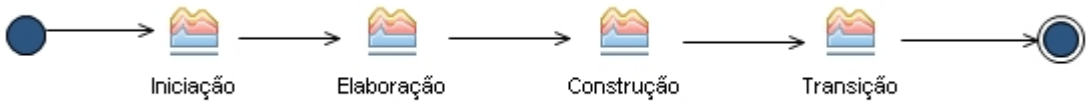
Artefatos – FasesProcess				
ID do artefato	**Inspiração**	**Metodologia de origem**	**FasesProcess**	**Justificativa**
A1	Documento de visão	OpenUP	Canvas	O modelo Canvas pode ser feito em sala de aula em 2 horas-aula e dará aos alunos uma visão geral das características do negócio do produto escolhido, compreendendo os elementos de um documento de visão
A2	Work items list	OpenUP	Requisitos cadastrados no Trello	Esse cadastro pode ser acompanhado de maneira fácil por todos os integrantes e pelo professor, proporcionando um gerenciamento de requisitos até o final do projeto
A3	Use-case Specification	OpenUP	Especificação de caso de uso de maior risco	Mesma estrutura mantida
A4	Classes Diagram	OpenUP	Diagrama de classes	Mesma estrutura mantida
A5	Backlog do sistema	Scrum	Cadastro de Issues no Git	O conjunto de Issues corresponderá ao todo do projeto a ser entregue no final da etapa de construção, esse cadastro servirá para que haja um gerenciamento a visão do desenvolvimento do sistema para todos os integrantes
A6	Backlog da Sprint	Scrum	Identificação das Issues a serem implementadas em cada Sprint	Essa identificação tornará visível e gerenciável a todos os integrantes o que deve ser feito na Sprint em execução e por quem deve ser feito, permitindo

				verificar o progresso do desenvolvimento
A7	Test Script	OpenUP	Relatório de testes	Mesma estrutura mantida, é uma forma simples e sistemática de registrar os resultados dos testes das funcionalidades implementadas em cada Sprint
A8	Backlog da Sprint	Scrum	Cadastro de Issues com resultados dos testes	Manutenção do controle do A5
A9	Build executável	Scrum	Versão do produto	Artefato gerado ao final de cada Sprint, é um resultado potencialmente entregável

Fonte: Elaborado pela autora (2017).

O processo foi incialmente definido e depois implementado no EPF Composer. Por fim, foi publicado[3] no site da FaSEs para que estivesse disponível a todos. Na Figura 3 está representado o fluxo do processo, que foi elaborado no EPF Composer. É um fluxo simples que define a ordem em que as fases do processo devem acontecer.

Figura 3 – Fluxo do FaSEs Process



Fonte: Elaborado pela autora (2017).

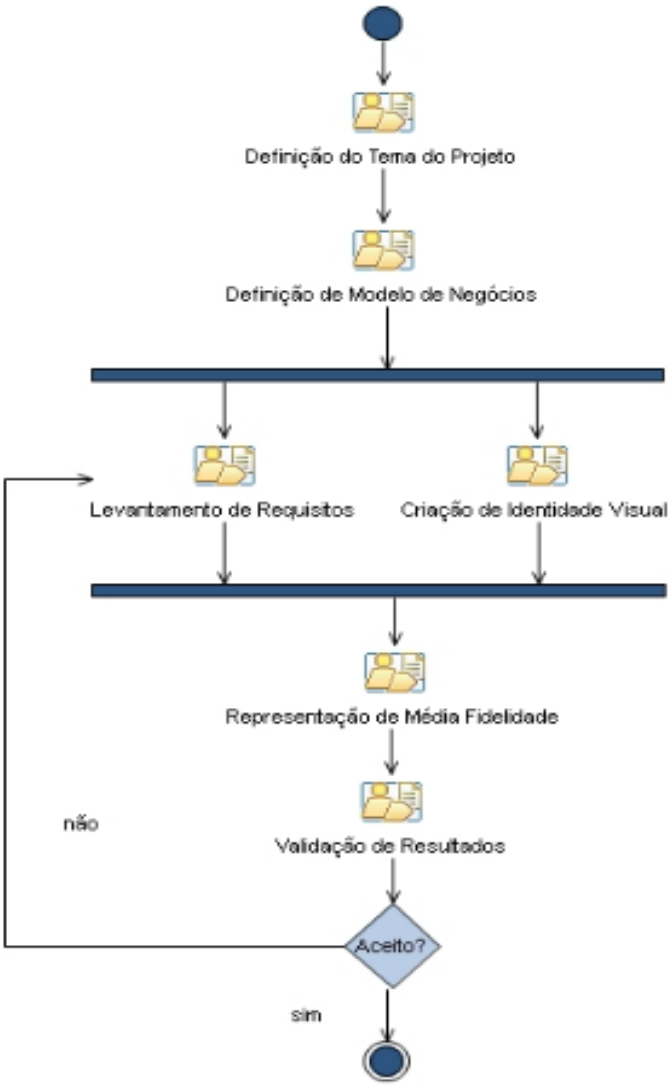
4.2. FASE INICIAÇÃO

Nessa fase do processo, *stakeholders* e toda a equipe trabalham juntos para definir o escopo do projeto, bem como os seus objetivos. Como já foi dito, a grande

[3] Disponível em: <http://fases.ifrn.edu.br/process/>

pergunta a ser respondida nessa fase é “O quê?”, ou seja, deve-se conceber e delimitar bem a ideia. Na Figura 4 está representado o diagrama de atividades da fase Iniciação, produzido no EPF Composer. Como pode ser observado no diagrama, a primeira atividade é Definição do Tema do Projeto, em seguida, a Descrição do Modelo de Negócios. Posteriormente, acontecem duas atividades paralelas: Levantamento de Requisitos e Criação de Identidade Visual e quando estas estivem completas, segue-se para a Representação de Médica Fidelidade. Por fim, os resultados devem ser validados e se houver necessidade de ajustes, retorna-se para a atividade de Levantamento de Requisitos e segue-se o fluxo.

Figura 4 – Diagrama de atividades da fase Iniciação



Fonte: Elaborado pela autora (2017).

No quadro 5 pode ser observada a estrutura da fase Iniciação dividida em Atividades, Tarefas, Artefatos e Papéis.

Quadro 5 – Estrutura da fase Iniciação

FASE INICIAÇÃO			
Atividade	**Tarefa**	**Artefato**	**Papéis**
Definição do tema do projeto	Capturar demanda da comunidade ou Continuar projeto existente ou Selecionar ideia a partir de brainstorming ou Selecionar ideia a partir de observação etnograficamente do Campus	Tema do projeto Relação de participantes da equipe	Analista Stakeholder Gerente
Definição de modelo de negócios	Participar ou replicar workshop de Canvas Produzir Canvas	Modelo de Canvas	Analista Stakeholder Gerente
Levantamento de requisitos	Realizar entrevistas e questionário Realizar análise de Benchmarking	Relatório das entrevistas Resultados do questionário Relatório de análise de benchmarking Relação de Requisitos	Analista Stakeholder
Criação de identidade visual	Definir identidade visual	Logomarca ou logotipo Paleta de cores Nome do projeto/release Business mockup	Designer
Representação de média fidelidade	Construir software mockup	Software mockup	Designer
Validação de resultados	Montar apresentação Apresentar resultados Avaliar resultados	Pitch Avaliação de resultados	Equipe Gerente Stakeholder

Fonte: Elaborado pela autora (2017).

A primeira atividade (definição do tema do projeto), tem quatro tarefas que excluem umas as outras, ou seja, deve ser escolhida apenas uma delas para ser executada; corresponde à origem da ideia do projeto. Como artefatos, as tarefas geram o Tema do projeto e a Relação de participantes da equipe. Os papéis envolvidos são os de analista, *stakeholder* e gerente.

A segunda atividade (definição de modelo de negócios) tem duas tarefas: Participar do Workshop de Canvas e Produzir Canvas. Na primeira tarefa, a equipe deve aprender os conceitos referentes ao Quadro de Modelo de Negócios (Canvas),

bem como aprender como produzi-lo. Na segunda tarefa, devem fazer o Canvas de fato, gerando como artefato, obviamente, o Modelo de Canvas do projeto. Analista, *stakeholder* e gerente realizam essas tarefas.

A terceira atividade (levantamento de requisitos) corresponde a duas tarefas: Realizar entrevistas e questionário e realizar análise de *benchmarking*. Essa atividade é muito importante pois é nela que serão definidos os requisitos do projeto, ou seja, aquilo que o projeto vai oferecer e entregar ao final do projeto. Todas as formas de levantamento de dados escolhidas (entrevistas, questionário e análise de benchmarking) têm o propósito de orientar a equipe no sentido do que já é produzido na área do projeto e do que os potenciais usuários esperam da ideia. Analista e *stakeholder* participam dessas tarefas e ao final são gerados Relatório das entrevistas, Resultados do questionário e Relatório de Benchmarking como artefatos, além da Relação de requisitos.

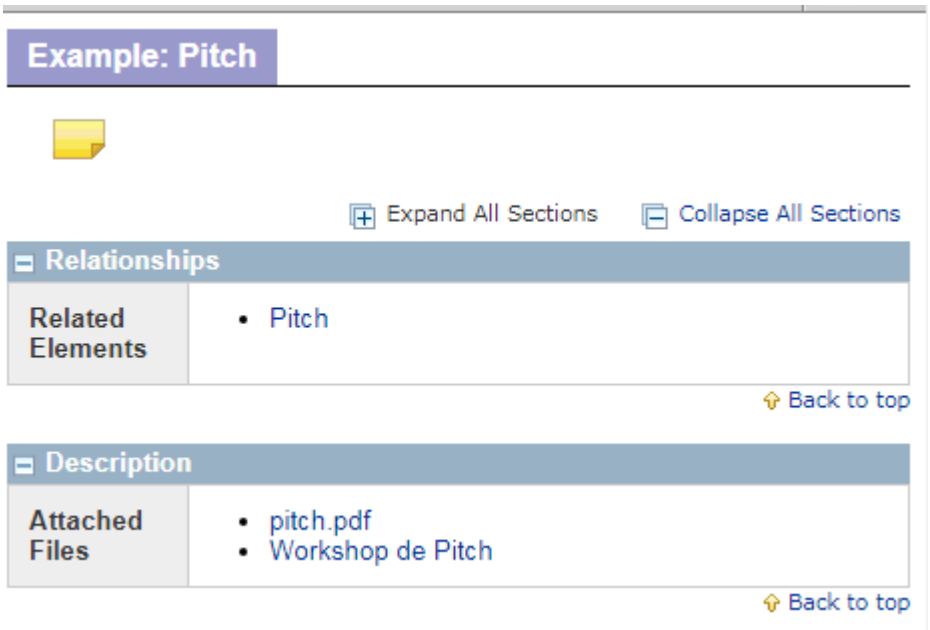
A quarta atividade (criação de identidade visual) só possui uma tarefa, que é desempenhada pelo *designer* e gera vários artefatos: logomarca ou logotipo, paleta de cores, nome do projeto e *Business Mockup*. Todos esses artefatos têm guias com exemplos e/ou indicações de ferramentas para serem utilizadas na sua produção.

A quinta atividade (representação de média fidelidade) também só possui uma tarefa: construir *software Mockup*. O *mockup* foi escolhido como representação de média fidelidade tendo em vista que prototipação demandaria muito tempo e recursos para ser feita, enquanto o *mockup* pode ser feito em menos tempo e servirá para validar a estrutura da informação de forma eficaz. Claramente, o artefato gerado é o Software *Mockup* e o *designer* da equipe é quem o faz.

A sexta atividade (Validação de Resultados) é dividida em três tarefas: Montar apresentação, desempenhada por todos os membros da equipe; Apresentar resultados, sob responsabilidade do gerente; e Avaliar resultados, o que deve ser feito pelo *stakeholder*. Nesse sentido, a equipe deve preparar uma apresentação que contemple os principais resultados da fase de Iniciação, com o intuito de conquistar a adesão e a concordância do *stakeholder*, que é o interessado no projeto. Os artefatos gerados são a Avaliação dos resultados e um *Pitch*, formato de apresentação persuasiva e rápida. O FaSEs Process possui um guia com um exemplo de *Pitch*,

além de uma apresentação de Workshop ensinando como montar um, como pode ser visto na Figura 5.

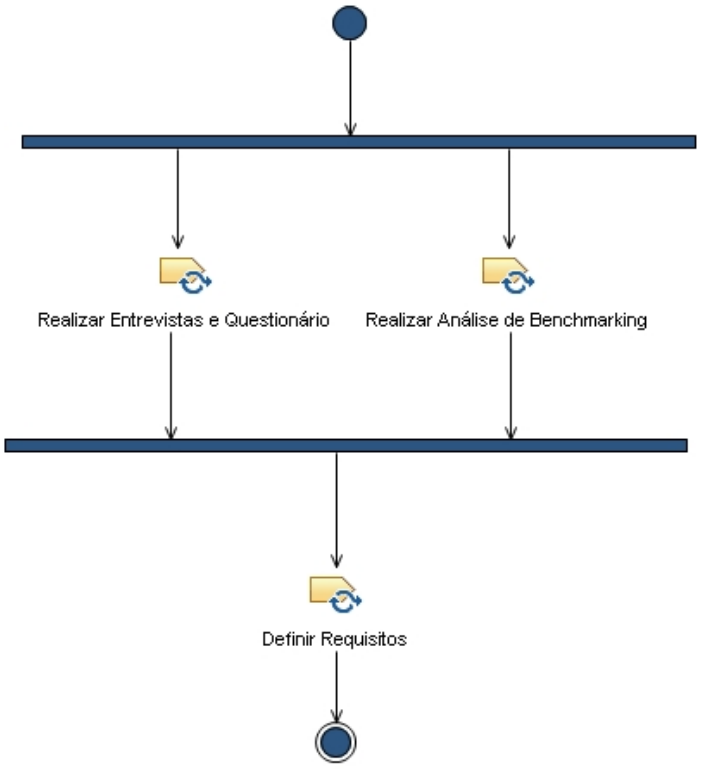
Figura 5 – Exemplo de guia



Fonte: Elaborado pela autora (2017).

Além dos diagramas de atividade de cada fase, também foram feitos diagramas para cada atividade, como pode ser visto na Figura 6, na qual está representado o diagrama da atividade Levantamento de Requisitos. Nesse diagrama está representado o fluxo das tarefas da atividade.

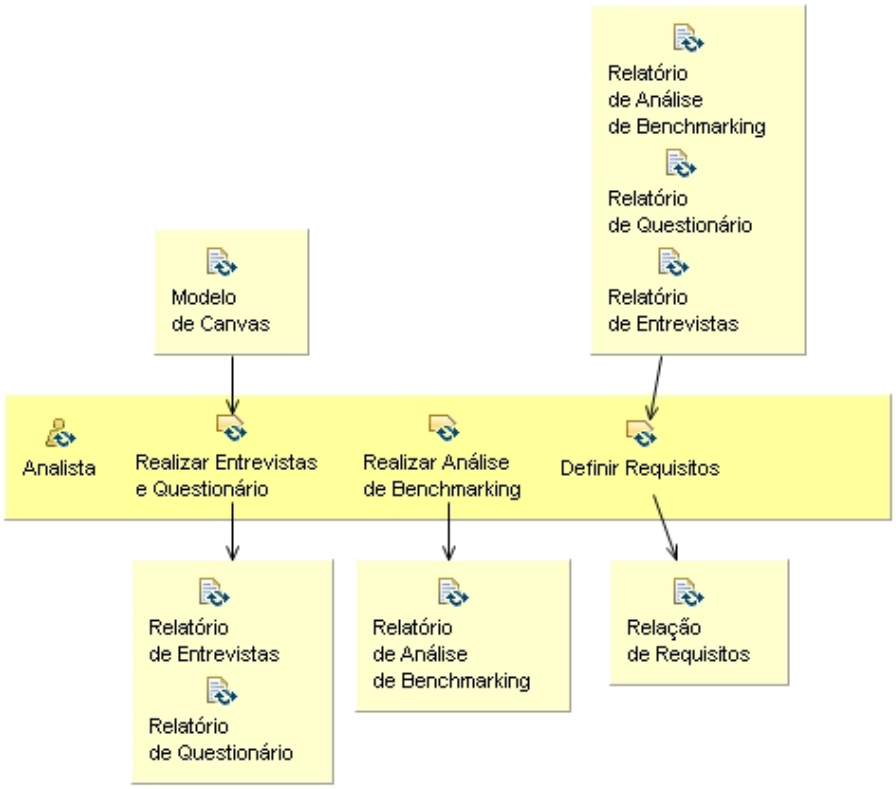
Figura 6 – Diagrama da atividade levantamento de requisitos



Fonte: Elaborado pela autora (2017).

Ademais, foi feita uma representação dos artefatos de entrada e de saída para cada tarefa, como pode ser visto na Figura 7. Essa representação também indica qual papel realiza as tarefas. Na representação abaixo, o analista da equipe faz as três tarefas, que precisam dos artefatos indicados na parte superior para que sejam realizadas e geram os artefatos indicados na parte inferior. Todas as atividades do processo têm o diagrama do tipo da Figura 6 e a representação do tipo da Figura 7.

Figura 7 – Artefatos de entrada e de saída de cada tarefa da fase de Levantamento de Requisitos

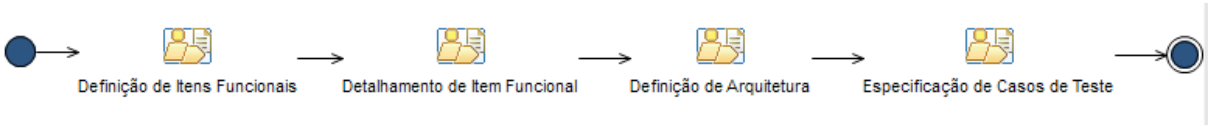


Fonte: Elaborado pela autora (2017).

4.3. FASE ELABORAÇÃO

Nessa fase a grande pergunta a ser respondida é: “como fazer?”. Agora que a ideia já está bem definida, precisa-se estabelecer como ela vai ser desenvolvida. É nessa fase que acontece detalhamento e especificação dos requisitos definidos na fase anterior, bem como o estabelecimento da base da arquitetura do processo. Na Figura 8, pode-se observar o digrama de atividade da fase de Elaboração, o qual estabelece uma sequência entre as quatro atividades: Definição de Itens Funcionais, Detalhamento de Item Funcional, Definição de Arquitetura e Especificação de Casos de Teste. Essas atividades devem ser feitas uma após a outra, na sequência indicada.

Figura 8 – Diagrama de atividades da fase Elaboração



Fonte: Elaborado pela autora (2017).

O quadro 6 mostra a estrutura da fase Elaboração, dividida em Atividades, Tarefas, Artefatos e Papéis.

Quadro 6 – Estrutura da fase Elaboração

FASE ELABORAÇÃO			
Atividade	**Tarefa**	**Artefato**	**Papéis**
Definição de itens funcionais	Definir itens funcionais do sistema	Repositório do Git com Issues cadastradas para cada item funcional	Arquiteto Desenvolvedor Gerente
Detalhamento de item funcional	Especificar item de maior risco	Documento de especificação de item de maior risco	Arquiteto Gerente
Definição de arquitetura	Desenvolver arquitetura	Diagrama de classes	Arquiteto Gerente
	Projetar banco de dados	Diagrama entidade-relacionamento	
Especificação de casos de teste	Especificar casos de teste	Documento de especificação de casos de teste	Tester

Fonte: Elaborado pela autora (2017).

A primeira atividade (definição de itens funcionais) só tem uma tarefa: definir itens funcionais do sistema. Essa tarefa é inspirada na metodologia *Scrum*, mais especificamente num elemento que esta metodologia denomina *Backlog do sistema* (PRIKLADNICKI; WILLI; MILANI, 2014). O *backlog do sistema* corresponde a uma lista de itens funcionais a serem implementados ao longo do desenvolvimento do projeto, ordenados por importância. Para a definição desses itens deve-se considerar os requisitos anteriormente definidos. Cada item será registrado como uma *issue* no repositório Git do projeto. Essa atividade deve ser realizada pelo arquiteto, pelo gerente e pelo desenvolvedor.

A segunda atividade (detalhamento de item funcional) tem como tarefa única Especificar Item de Maior Risco. Nessa tarefa será produzido um documento de Especificação do Item de Maior Risco, que deve obedecer ao *template* do processo. Esse documento deve possuir informações sobre: atores, pré-condições, pós-condições, fluxo básico, fluxo alternativo, fluxo de exceção, além do diagrama de atividades.

A terceira atividade (Definição de Arquitetura) é realizada pelo arquiteto e pelo gerente e tem duas tarefas: Desenvolver Arquitetura e Projetar Banco de Dados, que geram, respectivamente, os artefatos Diagrama de classes e Diagrama entidade-relacionamento. Apenas esses artefatos são gerados, tendo em vista que o processo busca atenuar a documentação resultante. Na Figura 9, pode ser visto que o artefato Diagrama de Classes tem um guia com um exemplo de diagrama de classes e com um tutorial de como produzir um.

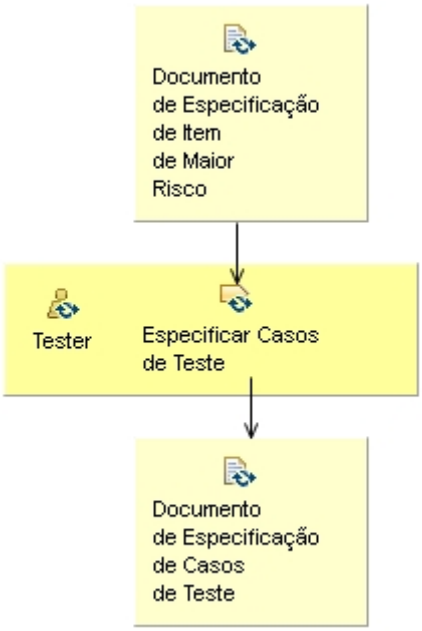
Figura 9 – Guia do artefato Diagrama de Classes

Fonte: Elaborado pela autora (2017).

A quarta atividade (Especificação de Casos de Teste) é realizada pelo *tester* e tem como única tarefa especificar casos de teste, que gera um Documento de especificação de casos de teste. Esse documento tem um guia no processo que indica

o modelo que ele deve seguir e depende do documento produzido na atividade anterior. A Figura 10 representa o detalhamento dessa quarta atividade.

Figura 10 – Detalhamento da atividade Especificação de Casos de Teste



Fonte: Elaborado pela autora (2017).

4.4. FASE CONSTRUÇÃO

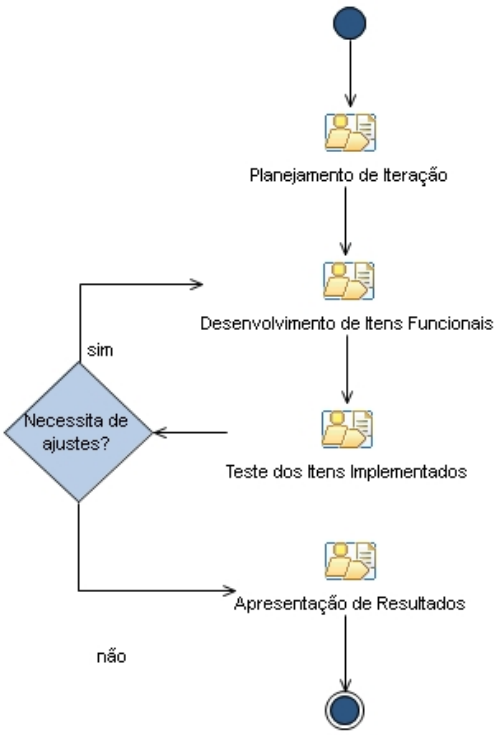
Nessa fase serão desenvolvidos os itens definidos na fase anterior, de acordo com a arquitetura definida. Como pode ser observado na Figura 11, a fase de Construção do processo é estruturada em uma iteração, que deve ser repetida quantas vezes for necessária. Ao final de cada iteração deve-se ter uma versão potencialmente executável do sistema.

Figura 11 – Fluxo da fase Construção

Fonte: Elaborado pela autora (2017).

Na Figura 12 está o diagrama de atividades da Iteração de construção. Primeiro, deve haver o Planejamento da Iteração, segue-se então para a atividade de Desenvolvimento de Itens Funcionais e depois é feito o Teste dos Itens Implementados e, caso sejam necessários ajustes, retorna-se para o Desenvolvimento de Itens Funcionais, até que não precise mais de ajustes e possa ser realizada a Apresentação de Resultados da iteração.

Figura 12 – Diagrama de atividades da Iteração da Construção



Fonte: Elaborado pela autora (2017).

A estrutura da fase Construção está no quadro 7, dividida em Atividades, Tarefas, Artefatos e Papéis.

Quadro 7 – Estrutura da fase Construção

FASE CONSTRUÇÃO			
Atividade	**Tarefa**	**Artefato**	**Papéis**
Planejamento da Iteração	Planejar iteração	Issues cadastras com a versão da iteração	Desenvolvedor Gerente
Desenvolvimento dos itens da Iteração	Desenvolver os itens da iteração	Versão executável do sistema	Desenvolvedor
Teste de itens implementados na iteração	Testar funcionalidades implementadas na iteração	Relatório de testes da iteração Issues de erro/bug	Tester
Apresentação de resultados	Apresentar resultados da iteração	Slides de apresentação de resultados	Gerente

Fonte: Elaborado pela autora (2017).

A primeira atividade (Planejamento da Iteração) é realizada pelo desenvolvedor e pelo gerente da equipe e tem apenas uma tarefa: planejar iteração. Esse planejamento corresponde a decidir quais dos itens funcionais definidos na fase anterior serão implementados na iteração da vez, levando em consideração a ordem de prioridade. Esse método também é inspirado na metodologia *Scrum*, na qual chama-se *backlog da Sprint* (PRIKLADNICKI; WILLI; MILANI, 2014). Como artefato, as *issues* cadastradas na fase anterior devem ser identificadas como pertencentes a atual iteração.

A segunda atividade (Desenvolvimento dos Itens da Iteração) também possui uma única tarefa: desenvolver os itens da iteração. Obviamente, essa tarefa é realizada pelos desenvolvedores da equipe. Ao final, deve-se ter uma versão potencialmente executável do sistema com os itens funcionais definidos para a iteração.

A terceira atividade (Teste de Itens Implementados) tem como única tarefa testar as funcionalidades implementadas na iteração e deve ser realizada pelo *tester* da equipe, considerando o Documento de Especificação de Casos de Teste. Ao final,

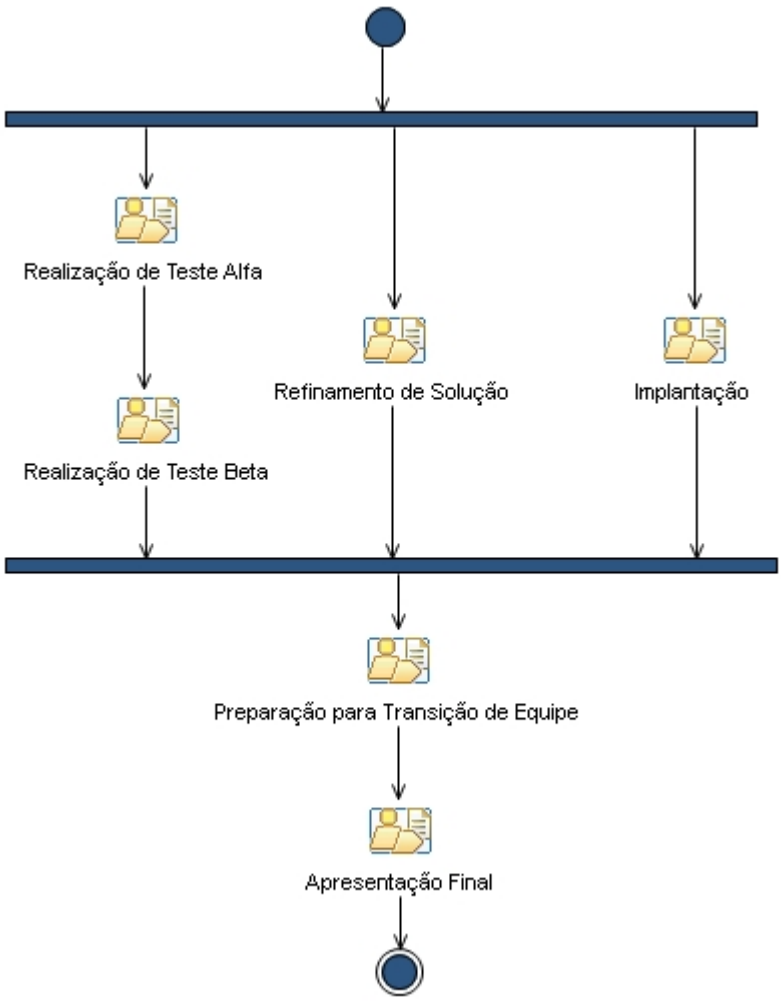
deve ser gerado um relatório de testes de acordo com o guia do processo e devem ser cadastradas *issues* de erro ou *bug* no repositório Git.

A quarta atividade (Apresentação de Resultados) tem apenas a tarefa apresentar resultados da iteração, que deve ser feita pelo gerente da equipe. Essa apresentação deve ser uma exposição em slides com o que foi prometido cumprir e o que de fato foi cumprido, bem como quais itens funcionais ainda precisam ser implementados, além de uma síntese dos testes realizados e erros encontrados. Também deve ser feita uma demonstração dos itens implementados.

4.5. FASE TRANSIÇÃO

A fase de Transição é a última fase do processo e seu objetivo é garantir que o software esteja pronto para ser entregue aos usuários finais. Como pode ser observado no diagrama da Figura 13, o Refinamento de Solução, ou seja, melhorias no desenvolvimento do software, ocorre em paralelo às realizações dos testes, pois os testes vão gerar novas necessidades ou ajustes no desenvolvimento. Primeiramente, ocorre a Realização de Teste Alfa e em seguida a Implantação do Sistema para que ocorra a Realização de Teste Beta. Quando os testes e o refinamento forem finalizados, o sistema deve ser implantado já com as melhorias e, em seguida, deve ser feita a Preparação para Transição de Equipe e, por fim, uma apresentação final.

Figura 13 – Diagrama de atividades da fase Transição



Fonte: Elaborado pela autora (2017).

A estrutura da fase Transição está no quadro 8, dividida em Atividades, Tarefas, Artefatos e Papéis.

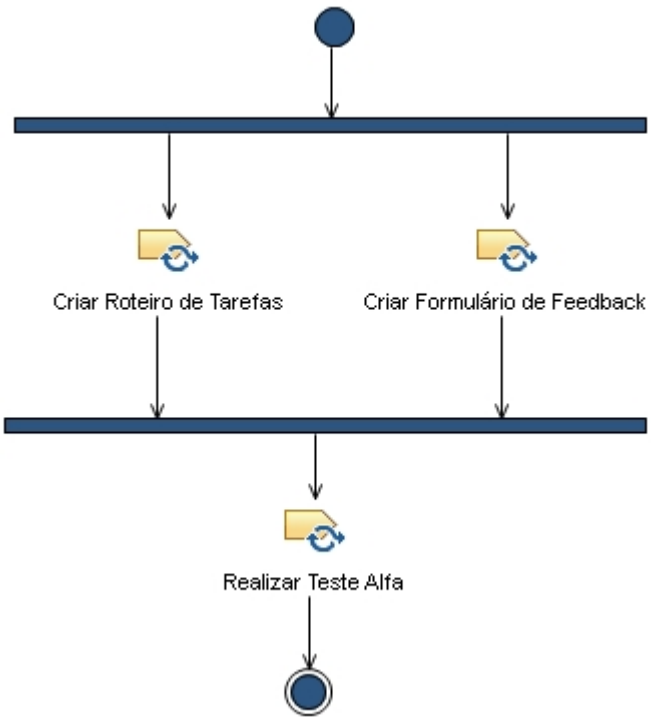
Quadro 8 – Estrutura da fase Transição

FASE TRANSIÇÃO			
Atividade	**Tarefa**	**Artefato**	**Papéis**
Realização de Teste Alfa	Criar roteiro de tarefas	Roteiro de tarefas	Tester
	Criar formulário de feedback	Formulário de feedback	Desenvolvedor
	Realizar teste alfa	Relatório de feedback Issues de bug ou melhoria	
Refinamento de Solução	Refinar solução	Versão executável	Desenvolvedor
Realização de Teste Beta	Criar formulário de feedback	Formulário de feedback	Desenvolvedor
	Realizar teste beta	Relatório de feedback Issues de bug ou melhoria	Stakeholder
Implantação	Aplicar identidade visual da FaSEs	Versão com ajustes da FaSEs	Desenvolvedor
	Hospedar sistema	Link do sistema hospedado	
	Alimentar sistema com dados reais	Sistema com dados reais	
Preparação para Transição de Equipe	Gerar script do banco	Script do banco	Desenvolvedor
	Preparar repositório	README Release do projeto	
Apresentação Final	Apresentar resultado final	Apresentação final	Gerente

Fonte: Elaborado pela autora (2017).

A primeira atividade (Realização de Teste Alfa) tem 3 tarefas: criar roteiro de tarefas, criar formulário de feedback e realizar teste alfa, como pode ser visto no diagrama da Figura 14. A primeira é feita pelo *tester* e a segunda e a terceira são realizadas pelos desenvolvedores.

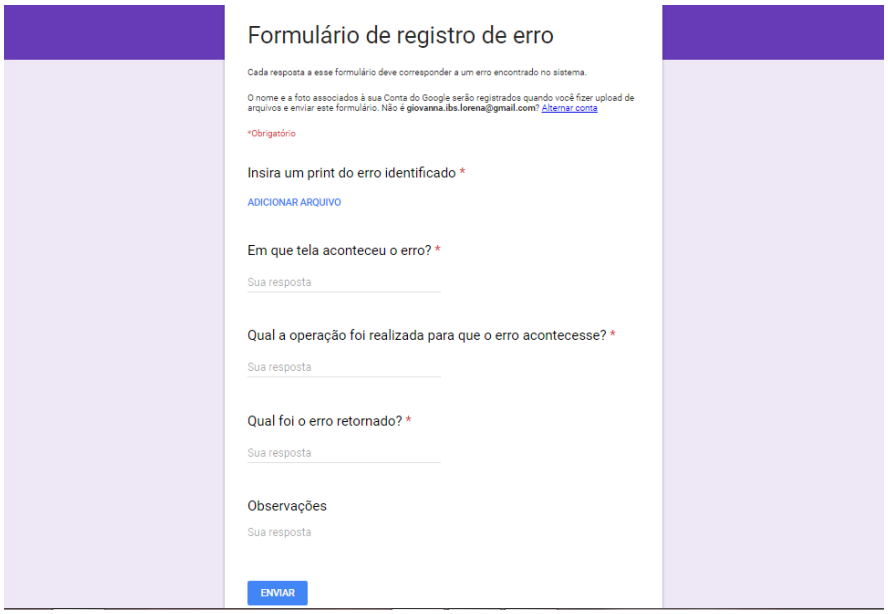
Figura 14 – Diagrama de atividades da atividade Realização de Teste Alfa



Fonte: Elaborado pela autora (2017).

As tarefas Criar Roteiro de Tarefas e Criar Formulário de Feedback acontecem paralelamente e geram, respectivamente, um roteiro de tarefas e um formulário de feedback como artefatos. Ambos artefatos possuem exemplos de guia para sua criação. O modelo do formulário de feedback pode ser visto na Figura 15. Ele é um formulário de registro de erro, ou seja, cada erro encontrado é reportado como uma resposta no formulário. Finalmente, a última tarefa é a própria Realizar Teste Alfa, que é feita pelos desenvolvedores – preferencialmente das outras equipes da turma, se for o caso. Esta última tarefa gera como artefato um relatório de feedback, que corresponde às respostas ao formulário de *feedback*, e *issues* de *bug* ou melhoria, de acordo com as respostas ao formulário.

Figura 15 – Modelo de formulário de feedback



Fonte: Elaborado pela autora (2017).

A segunda atividade (Refinamento de Solução) tem como tarefa única Refinar Solução, que é executada pelo desenvolvedor e gera uma versão executável do sistema. Antes mesmo dos testes, deve ser corrigido e finalizado aquilo que ficou em falta na fase de Construção e, ao longo dos testes, deve ser corrigido e/ou melhorado aquilo que os formulários de feedback indicaram.

A terceira atividade (Realização de Teste Beta) se assemelha muito à primeira, com exceção da tarefa de criar roteiro de tarefas e do papel que executa os testes, pois neste caso é o *stakeholder*, enquanto no Teste Alfa são os desenvolvedores. No Teste Beta, os potencias usuários utilizam o sistema sem um roteiro de tarefas pré-estabelecido, e podem também responder ao formulário de feedback para registrar erros encontrados.

A quarta atividade (Implantação) tem três tarefas, que são executadas pelo desenvolvedor: aplicar identidade visual da FaSEs, hospedar sistema e alimentar sistema com dados reais (Figura 16).

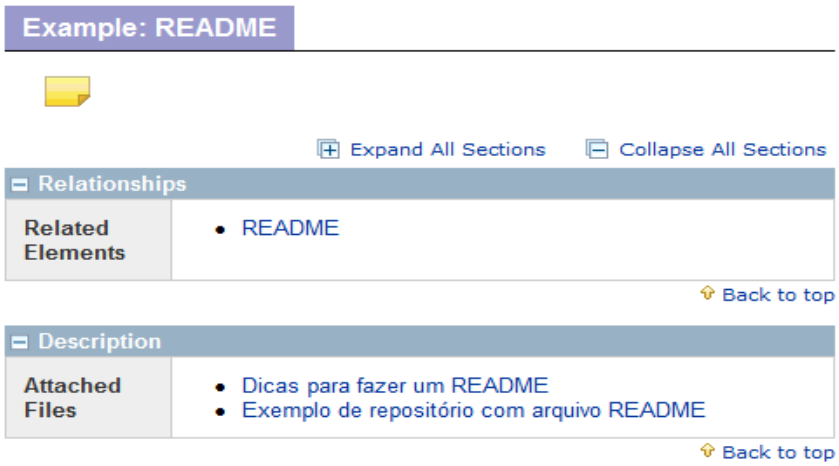
Figura 16 - Diagrama de atividades da atividade Implantação

Fonte: Elaborado pela autora (2017).

Na primeira tarefa, o logotipo da FaSEs e do IFRN-ZN devem ser incluídos no sistema, para indicar que o software foi desenvolvido em ambiente acadêmico. Em seguida, segue-se para a tarefa Hospedar Sistema e, por fim, Alimentar Sistema com Dados Reais. Essas tarefas geram como artefatos, respectivamente, versão com ajustes da FaSEs, link do sistema hospedado e sistema alimentado com dados reais.

A quinta atividade (Preparação para Transição de Equipe) tem duas tarefas, executadas pelos desenvolvedores: gerar script do banco e preparar repositório. Essa atividade servirá para que seja possível uma nova equipe continuar com o desenvolvimento do projeto. Para isso, são necessárias essas duas tarefas, que geram como artefatos o script do banco, um arquivo README e a *release* do projeto. Todos esses artefatos têm guias como orientações e exemplos. Na Figura 17 está o guia para o arquivo README, com orientações e um exemplo.

Figura 17 – Guia para arquivo README



Fonte: Elaborado pela autora (2017).

A sexta atividade (Apresentação Final) só possui a tarefa apresentar resultado final, que deve ser realizada pelo gerente da equipe. Essa apresentação deve ser uma síntese do que foi feito durante todo o processo, dando ênfase no que foi prometido a ser entregue e no que foi entregue no final. Além da apresentação dos resultados, deve ser feita uma demonstração do produto. O artefato gerado também tem um guia com exemplo de uma apresentação final.

5. **CONSIDERAÇÕES FINAIS**

Considerando-se a necessidade de um Processo de Desenvolvimento de Software especializado para o ambiente acadêmico do IFRN-ZN, que atendesse às suas peculiaridades, foi realizado o mapeamento sistemático com o intuito de buscar na literatura os processos de desenvolvimento de software para a academia. A maioria dos processos encontrados, entretanto, limitava-se ao desenvolvimento de jogos e/ou softwares educacionais, não se adequando à realidade aqui trabalhada. Nessa perspectiva, o FaSEs Process foi criado, com base em metodologias difundidas na área e considerando as condições do ambiente acadêmico – carga horária, nível de formação, recursos tecnológicos oferecidos pela instituição, etc. Além disso, o processo foi definido utilizando uma ferramenta padrão para definição de processos, o EPF *Composer*, e, por fim, tornado público para que seja usado academicamente.

Diante disso, propõe-se como trabalhos futuros a avaliação da utilização do processo nas turmas de 4º ano de Informática para Internet do Campus, bem como em projetos de pesquisa da área. Ademais, o processo pode ser refinado de acordo com o resultado da vivência e da avaliação de sua utilização. Os guias do FaSEs Process também podem ser atualizados com mais exemplos e *templates* e as descrições dos elementos do processo podem ser revisadas. Todos os trabalhos futuros devem ter o propósito de adequar cada vez mais o processo à realidade acadêmica, visando uma formação efetiva e eficaz, que prepare os discentes para o mercado de trabalho e para indústria.

REFERÊNCIAS

AMBLER, S.W. **Modelagem ágil**: práticas eficazes para a programação extrema e o processo unificado. [S. l.]: Bookman. 2004.

BATTISTELLA, Paulo Eduardo; VON WANGENHEIM, Christiane Gresse. ENgAGED: Um Processo de Desenvolvimento de Jogos para Ensinar Computação. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 380.

CAMPOS NETO, E. B. LOPES, A. S. B.; NASCIMENTO, D. S. C. Um relato de experiência da implantação de um modelo de fábrica de software escola (FaSEs). In: WEI - WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 25., 2017, São Paulo. **Anais...** São Paulo: Csbc, 2017. p. 2247-2256.

EPF Eclipse Process Framework Project. [S. l.: s. n.], 2017. Disponível em: <http://www.eclipse.org/epf>. Acesso em: nov. 2017.

FERNANDES, A. A.; TEIXEIRA, D. de S. **Fábrica de software**. São Paulo: Atlas, 2004.

GIBBS, W. **Software's chronic crisis:** scientific american. [S. l. : S. n.], 1994.

KITCHENHAM, B.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 26., 2004, Washington DC. **Anais…** Washington, 2004.

MALIK. **Software quality**: a practitioner's approach. [S. l.]: Tata McGraw-Hill Education, 2008.

MANIFESTO para o desenvolvimento ágil de software. [S. l.: s. n.], 2011. Disponível em:<http://www.manifestoagil.com.br/>. Acesso em: 08 nov. 2017.

MOURA, L. A. R. de. **Oficina com Alexander Osterwalder.** [S. l.]: Sebrae, 2014. Disponível em: <https://s3.amazonaws.com/academia.edu.documents/35803187/Memoria_Seminario_sobre_Canvas_Model_com_Alexander_Osterwlader_-_by_Luiz_Rolim.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1510162743&Signature=hOxvIOHLCc+xoOJ0XBKLmmKohng=&response-content-disposition=inline; filename=Memoria_Seminario_sobre_Canvas_Model_com.pdf>. Acesso em: 08 nov. 2017.

OSTERWALDER, A.; PIGNEUR, Y. **Business model generation:** inovação em modelos de negócios. [S. l.]: Alta Books, 2013.

PONTES, B. P.; ALEIXO, F.; MINORA, L. A. Processo acadêmico simplificado: uma proposta de processo para o CEFET-RN/DATINF. **HOLOS**, v. 3, p. 74-85, 2007.

PRESSMAN, R. S. **Engenharia de software:** uma abordagem profissional. 7. ed. São Paulo: AMGH, 2011.

PRIKLADNICKI, R.; WILLI, R.; MILANI, F. **Métodos ágeis para desenvolvimento de software**. [S. l.]: Bookman, 2014.

PMBOK project Managment Body of Knowledge Guide. [S. l.: s. n.], 2004.

RUP Rational Unified Process. [S. l.: s. n.]: 2017.Disponível em: <http://www.wthreex.com/rup/portugues/index.htm>. Acesso em: 12 ago. 2017.

SANTOS, R. E. et al. Ferramentas, métodos e experiências no ensino de engenharia de software: um mapeamento sistemático. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2014. **Anais...** 2014.

SCRUM Alliance. [S. l.: s. n.], 2017. Disponível em: <http://www.scrumalliance.org/>. Acesso em: 12 ago. 2017.

SCOTT, K. **O processo unificado explicado**. [S. l.]: Bookman, 2003.

SILVA, L. F.; LEITE, J. C. S. P.; BREITMAN, K. K. Ensino de engenharia de software: relato de experiências. In: WORKSHOP DE EDUCAÇÃO EM INFORMÁTICA, 12.,2004, Salvador. **Anais...** Salvador, 2004.

SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 27., 2016, Uberlândia. Anais... Uberlância, 2016.

SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p.

XP eXtreme Programming. [S. l.: s. n.], 2011. Disponível em: <http://www.extremeprogramming.org>. Acesso em: ago. 2017.