

**UNIVERSIDADE FEDERAL DO PAMPA**

**JESSICA CRISTINA BRASIL PEREIRA**

**UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE PARA/COM  
CRIANÇAS: INTEGRAÇÃO ENTRE ENGENHARIA DE SOFTWARE E  
INTERAÇÃO HUMANO-COMPUTADOR**

**Alegrete  
2016**

**JESSICA CRISTINA BRASIL PEREIRA**

**UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE PARA/COM  
CRIANÇAS: INTEGRAÇÃO ENTRE ENGENHARIA DE SOFTWARE E  
INTERAÇÃO HUMANO-COMPUTADOR**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Orientadora: Amanda Meincke Melo

**Alegrete  
2016**

**JESSICA CRISTINA BRASIL PEREIRA**

**UM PROCESSO DE DESENVOLVIMENTO DE SOFTWARE PARA/COM  
CRIANÇAS: INTEGRAÇÃO ENTRE ENGENHARIA DE SOFTWARE E  
INTERAÇÃO HUMANO-COMPUTADOR**

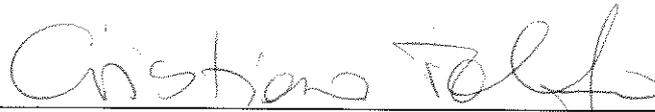
Trabalho de Conclusão de Curso  
apresentado ao Curso de Engenharia de  
Software da Universidade Federal do  
Pampa, como requisito parcial para  
obtenção do Título de Bacharel em  
Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em: 01 de dezembro de  
2016

Banca examinadora:

  
\_\_\_\_\_  
Prof.<sup>a</sup>. Dr.<sup>a</sup>. Amanda Meincke Melo  
Orientador  
UNIPAMPA

  
\_\_\_\_\_  
Prof. Me. Jean Felipe Patikowski Cheiran  
UNIPAMPA

  
\_\_\_\_\_  
Prof. Dr. Cristiano Tolfo  
UNIPAMPA

Dedico este trabalho a minha família e amigos.

## **AGRADECIMENTO**

Primeiramente agradeço a Deus pela vida, luz e pelo ar que respiro. Agradeço à orientadora do meu trabalho a Profa. Dr. Amanda Meincke Melo pelo apoio, confiança e conhecimentos compartilhados comigo. Agradeço também aos integrantes do projeto de pesquisa DICA-TDIC – Design Inclusivo com Crianças e Adolescentes de Tecnologias Digitais de Informação e Comunicação, que me proporcionaram oportunidades de conhecimento. Agradeço a minha família e aos meus amigos do curso pelo apoio e amizade.

“Deus nos fez perfeitos e não escolhe os capacitados, capacita os escolhidos. Fazer ou não fazer algo só depende de nossa vontade e perseverança”.

Albert Einstein

## RESUMO

Crianças são potenciais usuárias de tecnologias digitais tanto quanto os adultos. Há várias experiências relatadas na literatura do envolvimento da criança no desenvolvimento de software para seu uso. Entretanto, essas não apresentam uma abordagem de Engenharia de Software. Embora seja possível identificar diferentes trabalhos que integrem a área de Engenharia de Software e de Interação Humano-Computador, somente a área de Interação Humano-Computador tem mostrado contribuições significativas ao desenvolvimento de software para/com crianças. Este trabalho tem como objetivo, portanto, propor um processo de desenvolvimento de software para/com crianças, integrando métodos, técnicas e ferramentas de Engenharia de Software e Interação Humano-Computador, para auxiliar engenheiros de software e pesquisadores. Propõe-se, assim, a modelagem, em notação BPMN, de um processo de desenvolvimento de software para/com crianças, com auxílio de técnicas como observação participante, reuniões, entrevistas e análise documental. Como resultado do trabalho, tem-se um modelo de processo de desenvolvimento de software para/com crianças, resultante da análise de quatro estudos exploratórios e validado com a equipe do projeto de pesquisa DICA-TDIC – Design Inclusivo para/com Crianças e Adolescentes de Tecnologias Digitais de Informação e Comunicação. Este modelo pode ser adotado como referência para o desenvolvimento de software com crianças em novos projetos.

Palavras-Chave: Modelo de processo. BPMN. Crianças. Interação Humano-Computador. Engenharia de Software.

## **ABSTRACT**

Children are potential users of digital technologies as much as adults. There are several experiences described in the literature of children's involvement in software development to their use. However, these experiences do not show any Software Engineering approach. Although it is possible to identify works that integrate Human-Computer Interaction to Software Engineering, only the area of Human-Computer Interaction has shown significant contributions to the software development for/with children. This work aims to propose a software development process for/with children, integrating Software Engineering and Human-Computer Interaction methods, techniques and practices, to support researchers and software developers. In this way, we purpose modeling, with the BPMN notation, a process of software development for/with children, with the support of techniques such as participant observation, meetings, interviews and documental analysis. As the result of this work, there is a software development process model for/with children, resulting from the analysis of four exploratory studies and validated with the team from the research project Inclusive Design for/with Children and Teenagers of Communication and Information Digital Technologies (DICA-TDIC). This model can be adopted as a reference for the development of software with children in new projects.

Keywords: Process Model. BPMN. Children. Human-Computer Interaction. Software Engineering.

## LISTA DE FIGURAS

Figura 1 – Abordagem para o mapeamento e a modelagem de um processo.....	23
Figura 2 – Mapa mental de planejamento de construção de website.....	24
Figura 3 – Modelo de processo facilitador de atendimento .....	26
Figura 4 – Fluxograma do Método da Análise Documental.....	28
Figura 5 – Mapa mental do processo de Engenharia de Requisitos .....	31
Figura 6 – Modelo de Processo de Engenharia de Requisitos.....	33
Figura 7 – Modelo do subprocesso Levantar Requisitos.....	34
Figura 8 – Modelo do subprocesso Brainstorming .....	35
Figura 9 – Modelo do subprocesso Braindrawing .....	36
Figura 10 – Modelo do subprocesso Levantar histórias de usuário .....	37
Figura 11 – Modelo do subprocesso Analisar Requisitos.....	38
Figura 12 – Modelo do subprocesso Priorizar histórias de usuário .....	39
Figura 13 – Modelo do subprocesso Storyboarding .....	40
Figura 14 – Modelo do subprocesso Especificar Requisitos .....	41
Figura 15 – Modelo do subprocesso Validar Requisitos .....	42
Figura 16 – Modelo do subprocesso Validar histórias de usuário .....	43
Figura 17 – Mapa mental de o processo de Desenvolvimento de Software.....	44
Figura 18 – Modelo do processo Desenvolvimento de Software.....	46
Figura 19 – Modelo do subprocesso Engenharia de Requisitos atualizados .....	47
Figura 20 – Modelo do subprocesso Analisar Requisitos.....	48
Figura 21 – Modelo do subprocesso Desenvolvimento Iterativo .....	49
Figura 22 – Modelo do subprocesso Mockups.....	50
Figura 23 – Modelo do subprocesso Validar Software .....	51
Figura 24 – Exemplo de processo com dois atores.....	58
Figura 25 – Exemplo dum processo para cada ator.....	59
Figura 26 – Um processo para cada ator, ambos descritos.....	60
Figura 27 – Mapa mental da análise dos processos de desenvolvimento de software para/com crianças.....	64
Figura 28 – Mapa mental de proposta do processo de desenvolvimento de software .....	65
Figura 29 – Modelo do subprocesso Analogia Participativa.....	66
Figura 30 – Lista de histórias de usuário (LÔBO, 2016).....	69

## LISTA DE QUADROS

Quadro 1 – Trabalhos encontrados e suas contribuições .....	19
Quadro 2 – Elementos básicos do fluxo de processo da notação BPMN .....	25
Quadro 3 – Fluxo de Processo de Engenharia de Requisitos .....	32
Quadro 4 – Fluxo de o processo Desenvolvimento de Software.....	45
Quadro 5– Calendário no PROJOVEM-Adolescente (LÔBO, 2016).....	67
Quadro 6– Calendário na Moradia Transitória (LÔBO, 2016).....	68
Quadro 7– Tabela de Histórias de usuário (LÔBO, 2016).....	69

## LISTA DE ABREVIATURAS E SIGLAS

BPMN – *Business Process Model Notation*

DICA-TDIC – Design Inclusivo para/com Crianças e Adolescentes de Tecnologias Digitais de Informação e Comunicação

DI – Desenvolvimento Iterativo

DP – Design Participativo

DS – Desenvolvimento de Software

DUE – *Design Usability Evaluation*

ER – Engenharia de Requisitos

ES – Engenharia de Software

IDC – *Interaction Design and Children*

IHC – Interação Humano-Computador

IHC – Simpósio sobre Fatores Humanos em Sistemas Computacionais

MIT – *Model Inspection Techniques for Usability*

PATHY – *Persona Empathy*

PD4CAT – *Participatory Design Method for Customized Assistive Technology*

SBES – Simpósio Brasileiro de Engenharia de Software

Web DUE – *Web Design Usability Evaluation*

WE-QT – *Web Evaluation – Question Technique*

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>13</b>
<b>2 ESTADO DA ARTE .....</b>	<b>15</b>
2.1 Perspectiva da Engenharia de Software sobre a integração entre IHC e ES ..	15
2.2 Perspectiva da Interação Humano-Computador sobre a integração entre IHC e ES .....	17
2.3 Desenvolvimentos de tecnologia para/com crianças .....	18
2.4 Considerações finais do capítulo .....	19
<b>3 MATERIAIS E MÉTODOS .....</b>	<b>22</b>
3.1 Abordagem para o mapeamento e a modelagem de processo .....	22
3.2 Técnicas selecionadas.....	26
<b>4 RESULTADOS.....</b>	<b>30</b>
4.1 Um modelo de processo de Engenharia de Requisitos com a participação de crianças .....	30
4.2 Um modelo de processo de desenvolvimento de software com a participação de crianças .....	43
4.3 Considerações finais do capítulo .....	52
<b>5 CONSIDERAÇÕES FINAIS .....</b>	<b>53</b>
<b>REFERÊNCIAS.....</b>	<b>54</b>
<b>APÊNDICES .....</b>	<b>58</b>
<b>ANEXOS .....</b>	<b>67</b>

## 1 INTRODUÇÃO

As tecnologias digitais têm sido cada vez mais utilizadas pelas pessoas. Essa utilização impacta significativamente na forma como se comunicam, entre outros. Crianças<sup>1</sup>, como adultos, também desfrutam e fazem dessa utilização uma atividade diária.

As áreas Engenharia de Software (ES) e Interação Humano-Computador (IHC) possuem focos diferentes no desenvolvimento de software. Atualmente existem alternativas para a integração de ambas as áreas (BARBOSA; SILVA, 2010). Entretanto, apenas a área de IHC tem mostrado contribuições significativas ao desenvolvimento de software para/com crianças (BOSCARIOLI *et al.*, 2012; GRAY *et al.*, 2015; HANSEN; IVERSEN, 2013).

Processos e métodos da Engenharia de Software não consideram explicitamente o público infantil como possível parte interessada no desenvolvimento de software. Assim, não dão ouvidos às crianças, ao que elas querem ou porque querem e como querem. Dito isso, entende-se ser necessária uma abordagem sistemática de um processo de desenvolvimento de software para/com crianças para apoiar engenheiros de software no desenvolvimento específico para esse público, levando em conta aspectos próprios das crianças.

Para que seja possível o desenvolvimento de tecnologias para/com crianças, entretanto, é necessário insumos para auxiliar os profissionais de Engenharia de Software nesse processo, necessitando integrar princípios, processos e técnicas de IHC.

Este trabalho tem como objetivo geral, portanto, representar em notação BPMN um processo de desenvolvimento de software para/com crianças, integrando práticas da Engenharia de Software e da Interação Humano-Computador. Seus objetivos específicos são:

1. Compreender estratégias de integração entre Engenharia de Software e Interação Humano-Computador;
2. Documentar atividades e práticas de desenvolvimento de software no escopo do projeto de pesquisa DICA-TDIC – Design Inclusivo para/com

---

<sup>1</sup> Segundo a Convenção sobre os Direitos da Criança (UNICEF, 1989) toda pessoa que tem menos de 18 anos é considerada criança.

Crianças e Adolescentes de Tecnologias Digitais de Informação e Comunicação;

3. Orientar o desenvolvimento de software para/com crianças numa perspectiva de Engenharia de Software.

Inicialmente, realizou-se uma revisão de literatura com ênfase na integração entre Engenharia de Software e Interação Humano-Computador e também em como são desenvolvidas tecnologias para/com crianças. Além disso, o trabalho envolve a observação participante, a análise documental e a realização de entrevistas no contexto do projeto de pesquisa DICA-TDIC – Design Inclusivo para/com Crianças e Adolescentes de Tecnologias Digitais de Informação e Comunicação.

Tem-se como resultado uma representação, em notação BPMN (do inglês, *Business Process Model Notation*), do modelo de processo de desenvolvimento de software, abstraído a partir da observação de dois estudos exploratórios de Engenharia de Requisitos com a participação de crianças (LÔBO, 2016) e baseado em outros dois estudos exploratórios de desenvolvimento de software com o envolvimento de crianças, realizados por integrantes do projeto de pesquisa DICA-TDIC.

O texto deste trabalho está organizado da seguinte maneira. No Capítulo 2 é apresentado o estado da arte sobre a integração das áreas IHC e ES e do desenvolvimento de software para/com crianças. No Capítulo 3 estão descritos os materiais e métodos, as técnicas e as ferramentas utilizadas para o desenvolvimento do trabalho. No Capítulo 4 estão os resultados obtidos e no Capítulo 5 as conclusões juntamente com a indicação de possibilidades de trabalhos futuros.

## 2 ESTADO DA ARTE

Enquanto a Interação Humano-Computador (IHC) se preocupa com a concepção, a avaliação e a implementação de sistemas de computação interativos para o uso humano, juntamente com os principais fenômenos relacionados a esse uso (HEWETT *et al.*, 1992); a Engenharia de Software (ES), segundo Pressman (2006), é uma abordagem sistemática e disciplinada para o desenvolvimento de software.

Para Barbosa e Silva (2010), na IHC o uso do sistema deve guiar seu desenvolvimento, em contraste à abordagem da ES, que geralmente adota a perspectiva de *design* centrada no sistema. De acordo com os autores, existem basicamente três maneiras conhecidas de integração dessas duas áreas e são:

- Verificação de atividades de IHC que podem ser inseridas em processos de software;
- Verificação de processos de IHC que podem ser utilizados em paralelo a processos de ES;
- Verificação de características que se preocupam com qualidade de uso em um processo de software.

No contexto da produção de tecnologias digitais para/com crianças, em particular, o desenvolvimento de *software* envolve a participação de usuários e utiliza a abordagem do Design Participativo (DP), que Preece *et al.* (2005, p. 326) definem como os usuários projetando o produto com a equipe de *designers* envolvidos ativamente no desenvolvimento.

Este capítulo apresenta a pesquisa realizada em bases digitais, nas quais foram procurados trabalhos que estivessem no contexto da integração entre ES e IHC na produção de tecnologias digitais para/com crianças. Pesquisaram-se trabalhos nos anais dos eventos Simpósio Brasileiro de Engenharia de Software (SBES), Simpósio sobre Fatores Humanos em Sistemas Computacionais (IHC) e *Interaction Design and Children* (IDC).

### 2.1 Perspectiva da Engenharia de Software sobre a integração entre IHC e ES

Trabalhos publicados no Simpósio Brasileiro de Engenharia de Software (SBES) têm considerado a integração entre Engenharia de Software e Interação

Humano-Computador em diferentes perspectivas. Entretanto, não consideram o domínio de desenvolvimento de software para/com crianças.

Fernandes *et al.* (2012) propõem a técnica *Web Evaluation – Question Technique* (WE-QT), que é uma inspeção de usabilidade voltada a inspetores inexperientes. A técnica tem ênfase na avaliação de aplicativos *web* para a plataforma *mobile* e se baseia em questões que visam a reduzir as dificuldades dos inspetores em sua adoção.

Assim como Fernandes *et al.* (2012), Rivero *et al.* (2012) apresentam uma técnica de avaliação de usabilidade Web DUE (do inglês, *Web Design Usability Evaluation*) para ser integrada ao desenvolvimento de software, que visa a permitir a avaliação de protótipos ou *mockups* de baixa fidelidade durante o *design* de aplicativos *mobile*, utilizando um conjunto de itens de verificação de usabilidade para cada zona da página.

Rivero *et al.* (2014) apresentam um estudo empírico para avaliar a técnica DUE (do inglês, *Design Usability Evaluation*). Nessa avaliação é verificada se essa técnica realmente é capaz de ajudar as equipes de desenvolvimento na melhoria da qualidade das aplicações, utilizando-a em uma situação real e obtendo bons resultados sob o ponto de vista dos usuários. Já Valentim *et al.* (2014) apresentam a avaliação e a evolução da técnica MIT2 que faz parte de um conjunto de técnicas de inspeção para avaliações de usabilidade chamado MIT (do inglês, *Model Inspection Techniques for Usability Evaluation*). A técnica MIT2 inspeciona *mockups* e suas evoluções realizadas por engenheiros de software.

Cartaxo *et al.* (2013) apresentam um estudo empírico do impacto do Scrum na satisfação do cliente. Os resultados indicam que na amostra utilizada não foi encontrada nenhuma evidência que o uso do Scrum tenha implicações na satisfação do cliente. Esse artigo aborda a avaliação do método Scrum, mas não colabora à customização de um modelo ou método de desenvolvimento de software.

Ferreira *et al.* (2015), finalmente, apresentam a técnica PATHY (do inglês, *Persona Empathy*), voltada à criação de *personas* para apoiar no desenvolvimento de aplicações *web* para dispositivos móveis. Com essa técnica, a experiência de usuários prospectivos pode ser contemplada no desenvolvimento de software mesmo quando não é possível ter acesso direto ao usuário final, criando-se *personas* de potenciais usuários, diferentemente do que ocorre no contexto em que é desenvolvido este trabalho.

Embora seja possível identificar diferentes trabalhos que integrem ES e IHC, nenhum deles é dedicado ao domínio de desenvolvimento de software para/com crianças ou à modelagem de processo de desenvolvimento de software.

## **2.2 Perspectiva da Interação Humano-Computador sobre a integração entre IHC e ES**

O Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC) colabora à integração entre ES e IHC no desenvolvimento de software para crianças, com trabalhos que visam a observar requisitos como acessibilidade e usabilidade em projeto e avaliação de tecnologias computacionais interativas. A ênfase desses trabalhos, no entanto, não está no envolvimento da criança como parceira de *design*, onde elas podem contribuir diretamente no desenvolvimento de um produto expressando suas ideias, gostos e interesses de acordo com Lôbo (2016) *apud* Druin (2002).

Borges *et al.* (2012) apresentam um estudo realizado para projetar um dispositivo de comunicação para uma criança com paralisia cerebral voltado ao desenvolvimento de sua linguagem e habilidades de comunicação. Para isso, utilizam o PD4CAT (do inglês, *Participatory Design Method for Customized Assistive Technology*), que é um método de Design Participativo para o apoio ao desenvolvimento de Tecnologia Assistiva onde foi feito um estudo de caso com uma equipe de terapeutas que analisaram o como uma criança específica se comportava em sua vida e convivendo em sociedade. Embora envolva o Design Participativo, este apresenta a ênfase no desenvolvimento de recursos de Tecnologia Assistiva.

Como Borges *et al.* (2012), Braz *et al.* (2014) apresentam uma discussão sobre como expor aos terapeutas uma tecnologia diferente das que estão acostumados a utilizar em seu trabalho. A tecnologia proposta oferece a eles a autonomia na criação de atividades para o público com autismo de modo que sejam personalizadas de acordo com as necessidades de cada indivíduo. O autores trabalharam em *co-design* com os terapeutas, não com as crianças com autismo.

Boscarioli *et al.* (2012) apresentam os resultados preliminares sobre a avaliação de jogos educativos para crianças surdas estudantes dos anos iniciais do Ensino Fundamental. A avaliação tem o foco em aspectos de interação, como usabilidade, acessibilidade e comunicabilidade. Embora o trabalho trate de

usabilidade para jogos educativos para crianças, este apresenta a ênfase na avaliação e não no desenvolvimento de software.

### 2.3 Desenvolvimentos de tecnologia para/com crianças

Embora o evento *Interaction Design and Children* (IDC) apresente várias experiências do envolvimento da criança no desenvolvimento de tecnologias digitais, nenhuma adota uma abordagem de ES. Entre os trabalhos identificados, foram encontrados um estudo sobre crianças no Design Participativo, um projeto de tecnologia para crianças utilizando Design Centrado no Usuário e a proposta de princípios de *design* para o Desenho Universal.

Hansen e Iversen (2013) realizam um estudo de como os pesquisadores de Design Participativo motivam as crianças a se envolverem no processo de DP. Para tanto os autores fizeram um estudo de caso onde observaram como os pesquisadores motivaram as crianças a participar da concepção do processo. Dentre os resultados obtidos, foi percebido que a formação de pares foi bem aceita, a aprendizagem foi vista como forte motivadora e o *play* – que significa utilizar ferramentas que façam com que eles pensem como em um jogo – foi considerada interessante e motivacional. Esse trabalho colabora com a seleção de atividades motivadoras, que podem ser utilizadas no processo de desenvolvimento de software com crianças.

Gray *et al.* (2015) apresentam um jogo para celular chamado *BrainQuest*, que estimula a atividade física em crianças. Também abordam o processo utilizado, com Design Centrado no Usuário, envolvendo uma equipe de psicólogos, peritos em IHC, educadores físicos e crianças. Nesse trabalho as crianças começaram a participar após a ideia ser criada e os desenvolvedores utilizaram a abordagem de Design Centrado no Usuário integrando ES e IHC com a utilização de *mockups*. Entretanto, difere deste trabalho por não envolver as crianças em todo o processo de desenvolvimento.

Puccini *et al.* (2013) apresentam princípios de *design* multisensoriais para serem utilizados no Desenho Universal com crianças com necessidades especiais para maximizar suas habilidades metacognitivas, estruturas algorítmicas, entre outros, para a sua adequação e adaptação ao *design* tecnológico. Um dos princípios diz que um aplicativo ou tecnologia deveria permitir que as crianças alcançassem as

respostas corretas de várias maneiras, diferentemente da sala de aula onde é comum haver somente uma maneira de se encontrar a resposta correta. Esse entre outros princípios colaboram para o desenvolvimento de tecnologias para/com crianças, mas com ênfase em crianças com necessidades especiais, diferentemente do contexto deste trabalho.

## 2.4 Considerações finais do capítulo

A revisão de literatura foi realizada a partir de pesquisas nos anais dos últimos cinco anos dos eventos mencionados. No SBES foram encontrados cinco trabalhos que tratam de integração entre ES e IHC e um trabalho apresentando um estudo sobre gestão de processo. No evento IHC foram encontrados três trabalhos que apresentam formas de integração entre IHC e ES. Já no IDC foram encontrados três trabalhos que estudam e relatam o desenvolvimento de tecnologias para/com crianças. O Quadro 1, a seguir, sumariza os trabalhos encontrados separadamente por eventos e perspectiva de pesquisa, autores e contribuições.

Quadro 1– Trabalhos encontrados e suas contribuições

<b>Base</b>	<b>Autores</b>	<b>Contribuições</b>
Base: SBES  Perspectiva da ES para a integração IHC e ES	Fernandes <i>et al.</i> (2012) Rivero <i>et al.</i> (2012)	Técnicas de inspeção de interface de usuário para serem integradas ao processo de desenvolvimento de software.
	Rivero <i>et al.</i> (2014) Valentim <i>et al.</i> (2014)	Avaliação de técnicas de inspeção de interface de usuário integradas ao processo de desenvolvimento de software.
	Cartaxo <i>et al.</i> (2013)	Gestão de processo de desenvolvimento com Scrum e seu impacto na

		satisfação do cliente.
	Ferreira <i>et al.</i> (2015)	Técnica de criação de personas para obter experiências de potenciais usuários quando não se tem acesso ao usuário final.
Base: IHC  Perspectiva da IHC para a integração IHC e ES	Borges <i>et al.</i> (2012) Braz <i>et al.</i> (2014)	Acessibilidade e usabilidade como requisitos no desenvolvimento de tecnologias computacionais interativas.
	Boscarioli <i>et al.</i> (2012)	Avaliação da acessibilidade, usabilidade e comunicabilidade de jogos educativos para crianças.
Base: IDC  Desenvolvimento de Tecnologia para/com Crianças	Gray <i>et al.</i> (2015)	Participação de crianças no <i>design</i> de tecnologias para seu uso.
	Puccini <i>et al.</i> (2013)	Princípios de <i>design</i> para serem utilizados com crianças com necessidades especiais.
	Hansen e Iversen (2013)	Estudo sobre a motivação das crianças na participação de Design Participativo.

Fonte: Autoria própria

Dentre os trabalhos identificados, nenhum apresenta uma proposta de integração entre IHC-ES no desenvolvimento de tecnologias digitais para/com crianças. Gray *et al.* (2015), em particular, apresentam um estudo que se aproxima à proposta do presente trabalho, visto que descrevem o desenvolvimento de um jogo *mobile* utilizando o Design Centrado no Usuário com crianças.

### 3 MATERIAIS E MÉTODOS

Para documentar as atividades e práticas no escopo do projeto de pesquisa DICA-TDIC – Design Inclusivo para/com Crianças e Adolescentes de Tecnologias Digitais de Informação e Comunicação, foi realizada a modelagem de processo com a notação BPMN (do inglês, *Business Process Model Notation*). Para chegar ao modelo de processo, além da revisão de literatura, técnicas como observação participante, reuniões, entrevistas e análise documental contribuíram. Este capítulo apresenta os materiais e métodos adotados no desenvolvimento do trabalho.

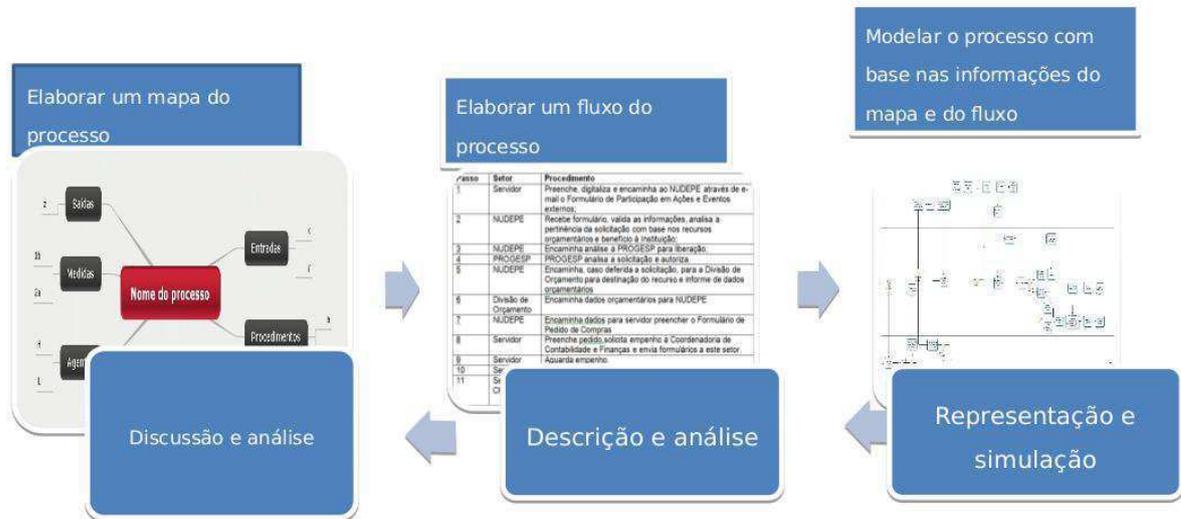
#### 3.1 Abordagem para o mapeamento e a modelagem de processo

A abordagem para o mapeamento e a modelagem do processo adotada é a proposta por Tolfo (2013) *apud* Segui (2015), organizada em três etapas:

1. Elaborar um mapa mental: neste mapa são descritas as principais atividades do processo, seus atores e suas entradas e saídas;
2. Elaborar fluxo de processo: as atividades descritas no mapa mental são organizadas em ordem de ocorrência no processo;
3. Modelar estado atual do processo: adota-se a notação *Business Process Model Notation* para representar o processo em análise.

Essas etapas geram um ciclo, como ilustrado na Figura 1. Ao término deste ciclo, os responsáveis pelo modelo refinam a modelagem, analisam e discutem o fluxo e o mapa do processo.

Figura 1 – Abordagem para o mapeamento e a modelagem de um processo



Fonte: Tolfo (2013) apud Segui (2015)

Esse ciclo foi realizado para cada processo adotado no contexto do projeto de pesquisa DICA-TDIC, iniciando pelo processo de Engenharia de Requisitos aplicado em dois estudos exploratórios (LÔBO, 2016). Em seguida, foi realizado para apoiar a abstração de outros dois processos de desenvolvimento de software envolvendo crianças.

### 3.1.1 Mapa mental

Mapa mental (BUZAN, 2005) é um diagrama visual que serve para expressar pensamentos ou conhecimentos sobre qualquer assunto, por exemplo, para produzir uma apresentação de aula, resumir um capítulo de um livro e/ou utilizar como agenda de atividades. A Figura 2 ilustra um exemplo de uso de mapa mental para orientar o projeto de um *website*.

Figura 2 – Mapa mental de planejamento de construção de um *website*



Fonte: Larrossa (2013)

No exemplo, pode-se notar uma informação central, que pode ser o título, e ramos onde são adicionadas informações sobre o assunto em forma de imagens e textos. Cada um desses ramos pode ser seguido por inúmeros outros ramos e, assim, sucessivamente.

Neste trabalho, para a elaboração de mapas mentais, foi utilizada a ferramenta *online* Mindomo<sup>2</sup> em sua versão aberta para estudantes.

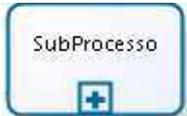
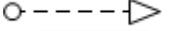
### 3.1.2 Business Process Modeling Notation (BPMN)

Almeida (2013) explica que BPMN (do inglês, *Business Process Model Notation*)<sup>3</sup> é uma notação de modelagem de processos de trabalho criada para prover uma notação gráfica padronizada e de fácil entendimento para processos de negócios, que fossem entendidas por todos os envolvidos, desde o analista de negócio aos programadores. O Quadro 2 apresenta os elementos básicos do fluxo de processo da notação com os respectivos significados.

<sup>2</sup> A ferramenta está disponível em: <<https://www.mindomo.com/pt/>>. Acesso em: 11/11/2016.

<sup>3</sup> O site da *Business Process Model Notation* está disponível em : <<http://www.bpmn.org/>>. Acesso em 13/11/2016.

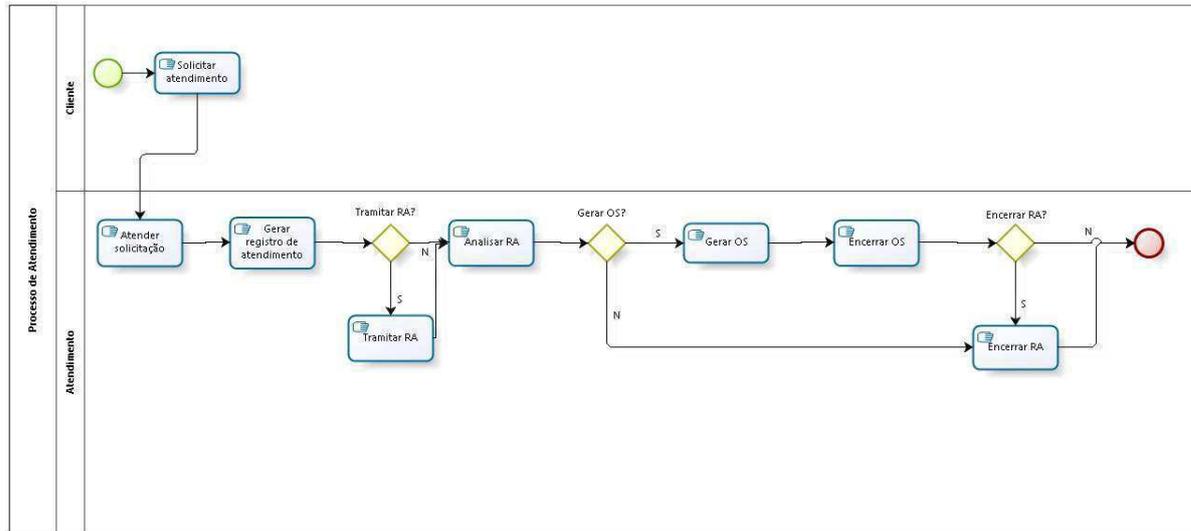
Quadro 2 – Elementos básicos do fluxo de processo da notação BPMN

<b>Categoria</b>	<b>Elemento</b>	<b>Descrição</b>
Atividades		Quadro que apresenta uma atividade ou passo do processo.
		Quadro que apresenta um subprocesso, ou seja, um processo que é feito para que aquela atividade seja cumprida.
Eventos		Evento do início do processo.
		Evento do final do processo.
Gateways		O <i>gateway</i> exclusivo é utilizado para pontos de decisões, de divergências e convergências no processo onde somente um caminho é seguido.
		O <i>gateway</i> paralelo é utilizado em pontos de convergência e um ou mais caminhos podem ser seguidos.
Conectores		Este conector é utilizado para dar ordem as atividades.
		Este conector é utilizado para associar dados e informações a objetos do processo.
		Este conector é utilizado para a comunicação entre processos.
Objetos de Dados		Proveem informações sobre as entradas e saídas de uma atividade.
Swimlanes		Partição que contém o processo.
		Partição horizontal dentro do processo utilizado para separar a pessoa que está realizando a atividade.
		Partições na vertical utilizadas para criar fases no processo.

Fonte: Almeida (2013)

A Figura 3 apresenta um exemplo de processo modelado na notação BPMN. O modelo descreve um processo de atendimento, que é iniciado pelo cliente solicitando atendimento, prosseguido por algumas atividades como, por exemplo, solicitar atendimento feito pelo cliente. A partir da atividade atender solicitação, o processo é continuado na raia de atendimento, encerrando-se com o evento final.

Figura 3 – Modelo de processo facilitador de atendimento



Fonte: Sarmento (2016)

A ferramenta Bizagi Modeler, que é gratuita, foi adotada para o desenvolvimento dos modelos na notação BPMN.

Para valorizar a participação das crianças no processo de desenvolvimento, foram experimentadas alternativas de representação e discutidas com os integrantes do projeto de pesquisa DICA-TDIC e um docente com notório saber em modelagem de processos. Foram experimentadas três alternativas que encontram-se nos APÊNDICES A, B e C.

### 3.2 Técnicas selecionadas

Para colaborar à prospecção de dados para a criação do modelo de processo, adotaram-se as seguintes técnicas: observação participante, reuniões, entrevistas e análise documental.

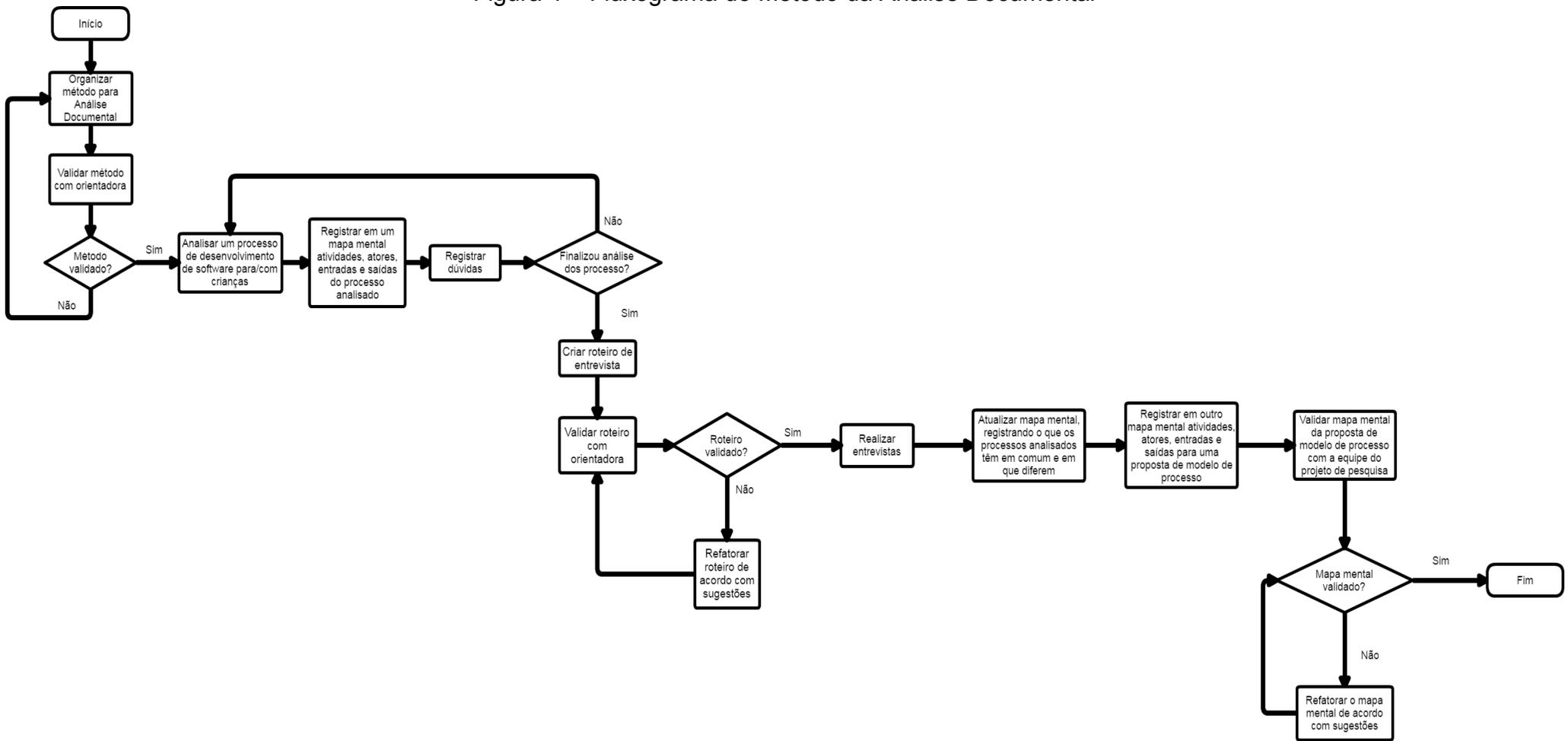
A observação participante, que Appolinário (2004, p. 144) define como “Processo de observação no qual o pesquisador participa ativamente como membro do grupo que ele está estudando, utilizando esta posição privilegiada para obter informações acerca desse grupo”, foi utilizada no contexto do projeto de pesquisa DICA-TDIC. Envolveu a participação em dois estudos exploratórios de Engenharia de Requisitos com crianças e reuniões semanais do projeto de pesquisa.

Além da participação em reuniões semanais da equipe do projeto de pesquisa DICA-TDIC, foram realizadas reuniões com os colegas do grupo que produziam tecnologias com crianças, visando a conhecer seus processos de trabalho. Essas reuniões colaboraram à dissolução de dúvidas, tomadas de decisões e à validação de modelos.

Foram realizadas entrevistas, segundo Barbosa e Silva (2010 p. 145), caracterizada como “uma das técnicas mais utilizadas de coleta de dados e levantamento de requisitos. Tratando-se de uma conversa guiada por um roteiro de perguntas ou tópicos.” Entrevistas foram organizadas visando a orientar as reuniões para conhecer os processos de software utilizados pelos colegas do grupo DICA-TDIC. A entrevista utilizada foi a não estruturada, que é uma entrevista flexível, onde são realizadas perguntas abertas para se aprofundar nos pontos desejados. Os roteiros das entrevistas encontram-se nos APÊNDICES D, E e F.

A técnica de análise documental, de acordo com Marconi e Lakatos (2009 p. 29), consiste na representação condensada da informação do conteúdo de um documento. Envolveu a verificação das agendas de trabalho de dois estudos exploratórios do projeto de pesquisa DICA-TDIC e a análise de dois processos de desenvolvimento de software para/com crianças descritos em projetos de Trabalho de Conclusão de Curso (TCC). A análise documental sobre a descrição dos processos de desenvolvimento de software está descrita em um fluxograma representado na Figura 4.

Figura 4 – Fluxograma do Método da Análise Documental



Fonte: Autoria própria

Conforme o fluxograma da Figura 4 apresenta, após validar o método de análise documental, foram analisados dois processos de desenvolvimento de software para/com crianças dos estudos exploratórios em andamento, documentados em dois projetos de TCC. Para isso, os textos dos projetos foram lidos em separado e, para cada um, foram registradas em um mapa mental (APÊNDICE G)<sup>4</sup> informações como etapas, atores, entradas e saídas presentes nos processos documentados. As dúvidas que surgiram também foram registradas. A partir das dúvidas registradas, criaram-se roteiros de entrevistas (APÊNDICES D, E e F).

Após as entrevistas, o mapa mental da análise dos processos foi atualizado com novos tópicos contendo os pontos comuns e diferentes entre os dois processos. O mapa mental foi, então, validado junto à equipe do projeto de pesquisa e orientadora.

Uma vez realizada a análise do mapa mental com o registro de informações sobre os processos, seus pontos em comum e diferenças, criou-se um novo mapa mental (APÊNDICE H) que representasse uma proposta única de modelo de processo de desenvolvimento de software para/com crianças. Este mapa mental foi validado por integrantes do projeto de pesquisa e orientadora, sendo atualizado conforme sugestões. O mapa mental do desenvolvimento de software está descrito na Figura 17 da Seção 4.2 e subsidiou a modelagem do processo de desenvolvimento para/com crianças.

---

<sup>4</sup> Mapa mental completo disponível em: <<https://www.mindomo.com/pt/mindmap/98170989d7484ad6adc6bc15b085c18d>>. Acesso em: 13/11/2016.

## 4 RESULTADOS

Como resultados, tem-se documentado em notação BPMN um processo de Engenharia de Requisitos (ER) e um processo de desenvolvimento de software. Ambos contemplam a participação de crianças.

### 4.1 Um modelo de processo de Engenharia de Requisitos com a participação de crianças

Para chegar ao modelo de processo de Engenharia de Requisitos (ER) em notação BPMN, realizaram-se a observação participante de dois estudos exploratórios envolvendo a produção de tecnologias digitais com crianças (LÔBO, 2016); a análise documental, envolvendo a verificação das agendas de trabalho de dois estudos exploratórios de Engenharia de Requisitos, que podem ser encontradas nos ANEXOS A e B, além da participação em reuniões semanais do projeto de pesquisa no período de 20/10/2015 a 01/11/2016.

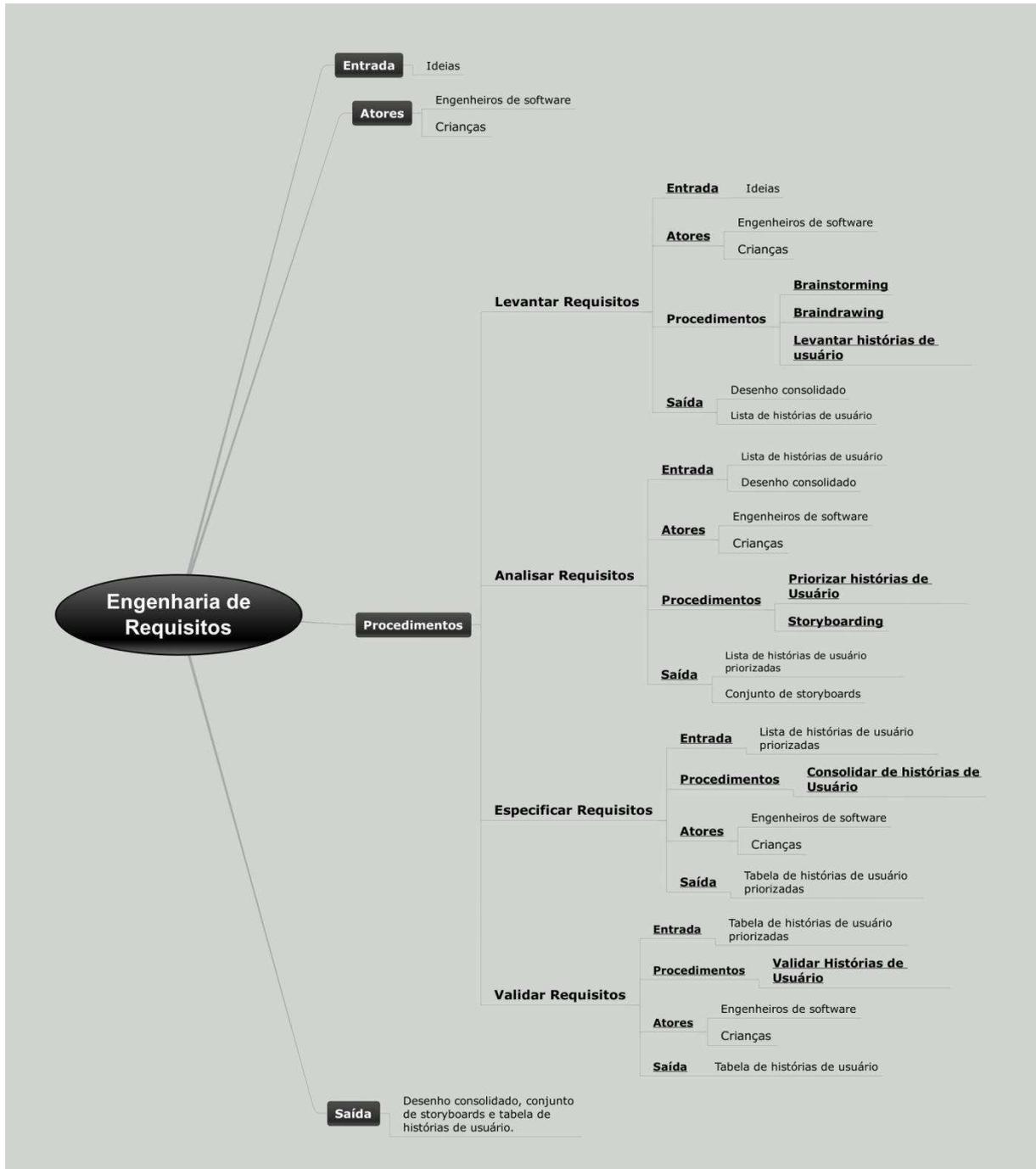
O modelo do processo de Engenharia de Requisitos foi inicialmente documentado através da abordagem de modelagem de processo apresentada em Tolfo (2013) *apud* Segui (2015) no período de 13/05/2016 a 10/06/2016. Também envolveu a validação em reuniões com a coordenadora do projeto de pesquisa DICA-TDIC, além de reunião com toda a equipe do projeto DICA-TDIC no dia 01/11/2016.

O mapa mental ilustrado na Figura 5<sup>5</sup> organiza o processo ER em entradas, que são ideias de software; atores, que são os engenheiros de software e as crianças; quatro procedimentos, que são levantar requisitos, analisar requisitos, especificar requisitos e validar requisitos, descritos como novos processos e detalhados no modelo de processo em um nível de abstração mais baixo; por fim, as saídas do processo, que são o desenho consolidado de um protótipo de baixa fidelidade da interface de usuário, um conjunto de *storyboards* e a tabela de histórias de usuário priorizadas. Enquanto as *storyboards* (MEDEIROS *et al.*, 2004) descrevem interações do usuário final com o sistema de software, as histórias de usuário (COHN, 2014) descrevem as principais funcionalidades do sistema na perspectiva do usuário.

---

<sup>5</sup> Mapa mental completo disponível em: < <https://www.mindomo.com/mindmap/mapa-mental-do-processo-de-er-atualizado-cc0dffa1aab54fb1bcb05d9dad222a3a>> Acesso em: 20/10/2016

Figura 5 – Mapa mental do processo de Engenharia de Requisitos



Fonte: Autoria própria

Após a modelagem do mapa mental, os procedimentos foram organizados em ordem de execução, conforme Quadro 3. Este quadro apresenta o fluxo dos procedimentos enumerados em ordem de acontecimento, com o setor descrevendo os atores que são os engenheiros de software e as crianças, já evidenciando a participação das crianças em cada etapa da Engenharia de Requisitos. O fluxo é

iniciado no passo 1, onde é realizado o Levantamento de Requisitos. No passo 2 acontece a Análise de Requisitos. Já, no passo 3, ocorre a Especificação de Requisitos, que diferentemente dos demais, é realizada somente pelos Engenheiros de Software. Por fim, no passo 4, ocorre a Validação de Requisitos.

Quadro 3 – Fluxo de Processo de Engenharia de Requisitos

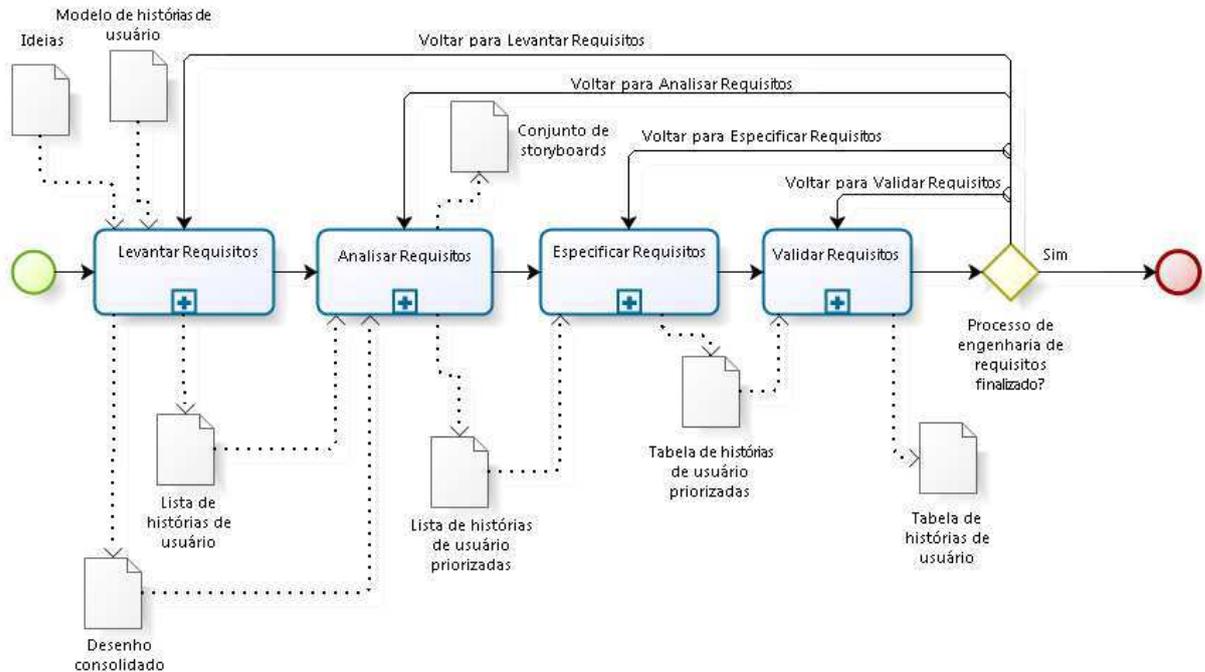
<b>Fluxo de Processo Engenharia de Requisitos</b>		
<b>Passo</b>	<b>Setor</b>	<b>Procedimento</b>
1	Engenheiros de Software	Levantar Requisitos
	Crianças	
2	Engenheiros de Software	Analisar Requisitos
	Crianças	
3	Engenheiros de Software	Especificar Requisitos
4	Engenheiros de Software	Validar Requisitos
	Crianças	

Fonte: Autoria Própria

A Figura 6 representa o nível de abstração mais alto do processo ER, organizando os procedimentos documentados no Quadro 3 com o ciclo baseado em Sommerville (2007). Esses procedimentos, em notação BPMN, são denominados subprocessos. O processo ER apresenta os subprocessos Levantar Requisitos, Analisar Requisitos, Especificar Requisitos e Validar Requisitos. Como entrada do processo Levantar Requisitos, têm-se ideias para o desenvolvimento de uma tecnologia com as crianças e um modelo de histórias de usuário. Como saída, tem-se um desenho consolidado, que representa um esboço preliminar da interface de usuário, e uma lista de histórias de usuários, que representam os requisitos funcionais a serem desenvolvidos. Estes são entrada para o subprocesso Analisar Requisitos. Como saídas do processo Analisar Requisitos, têm-se a lista de histórias de usuário priorizadas e um conjunto de *storyboards*, que descrevem algumas das interações do usuário com o sistema. A lista de histórias de usuário priorizadas serve de entrada para o subprocesso Especificar Requisitos, que apresenta como saída uma tabela de histórias de usuário priorizadas. Estas, finalmente, são entradas do subprocesso Validar Requisitos, que gera como saída uma tabela de histórias de

usuário. Antes de finalizar o processo, ocorre um evento de decisão, que determina se o processo de Engenharia de Requisitos foi efetivamente finalizado ou se ainda é necessário retornar a algum dos subprocessos de ER.

Figura 6 – Modelo de Processo de Engenharia de Requisitos



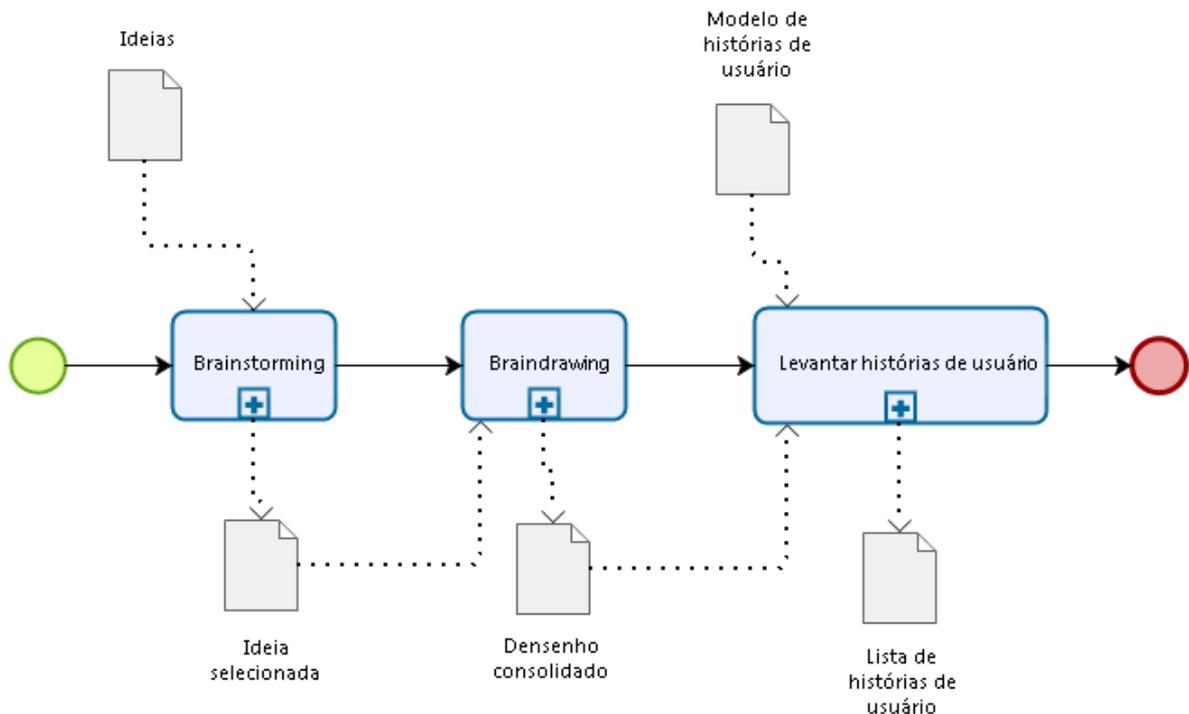
Fonte: Autoria própria

Cada subprocesso da ER é descrito em detalhes nas próximas subseções.

#### 4.1.1 Levantar Requisitos

A Figura 7 descreve o subprocesso Levantar Requisitos, que organiza as atividades Brainstorming, Braindrawing e Levantar histórias de usuário. Inicialmente, o Brainstorming recebe como entrada as ideias dos engenheiros de software de criar uma tecnologia digital com as crianças, gerando como saída uma ideia selecionada pelo grupo envolvido para o desenvolvimento de software. Esta ideia serve de insumo para o Braindrawing, que tem como saída um desenho consolidado, em baixa fidelidade, da interface de usuário ou parte dela. Este desenho serve de insumo para a atividade Levantar histórias de usuário, que também recebe como entrada um modelo de histórias de usuário e apresenta como saída uma lista de histórias de usuário.

Figura 7 – Modelo do subprocesso Levantar Requisitos

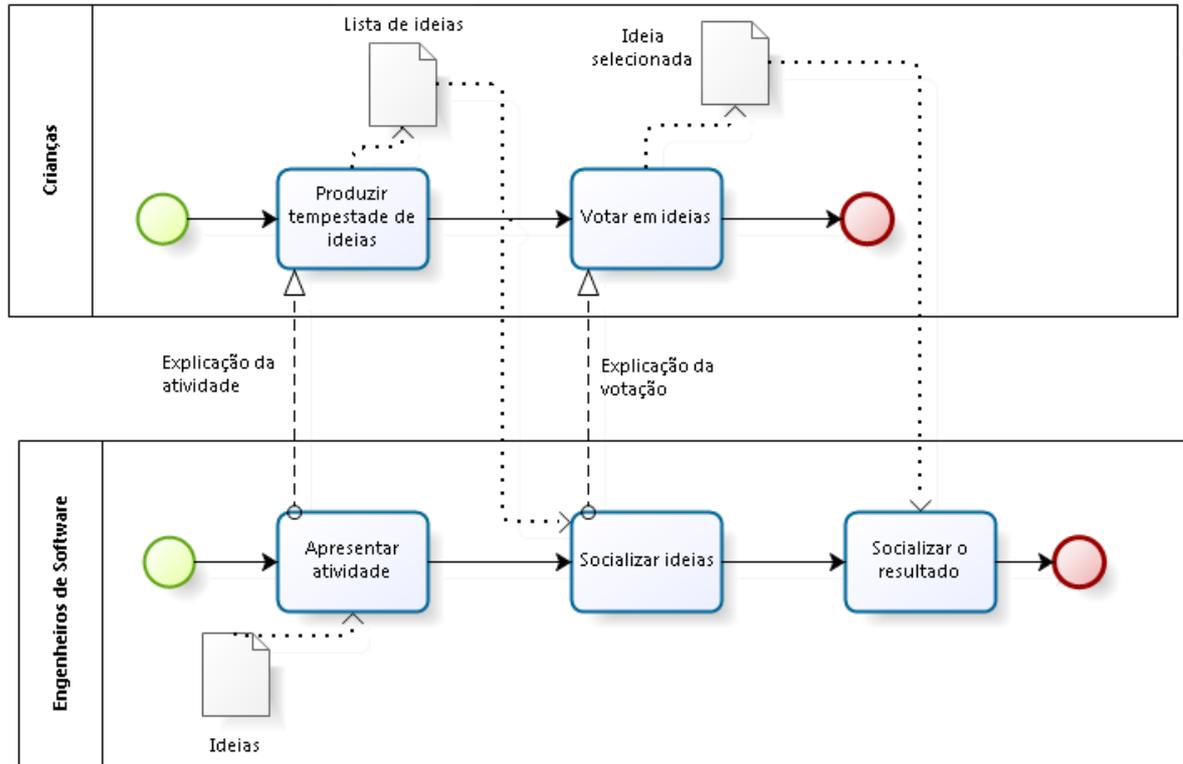


Fonte: Autoria própria

A participação das crianças no subprocesso Levantar Requisitos é explicitada a partir do detalhamento de cada um dos subprocessos apresentados na Figura 6.

No Brainstorming (Figura 8) (CARVALHO; CHIOSSI, 2001), os engenheiros de software, a partir da ideia inicial de desenvolver software com crianças, apresentam a atividade para as crianças, que produzem uma tempestade de ideias, gerando uma lista de ideias com possibilidades para o desenvolvimento de software. Os engenheiros de software, então, socializam as ideias geradas, explicando a votação para a escolha de uma ideia. As crianças prosseguem na escolha de uma ideia através de um processo de votação. Por fim, os engenheiros de software socializam o resultado da votação, gerando como saída a ideia selecionada.

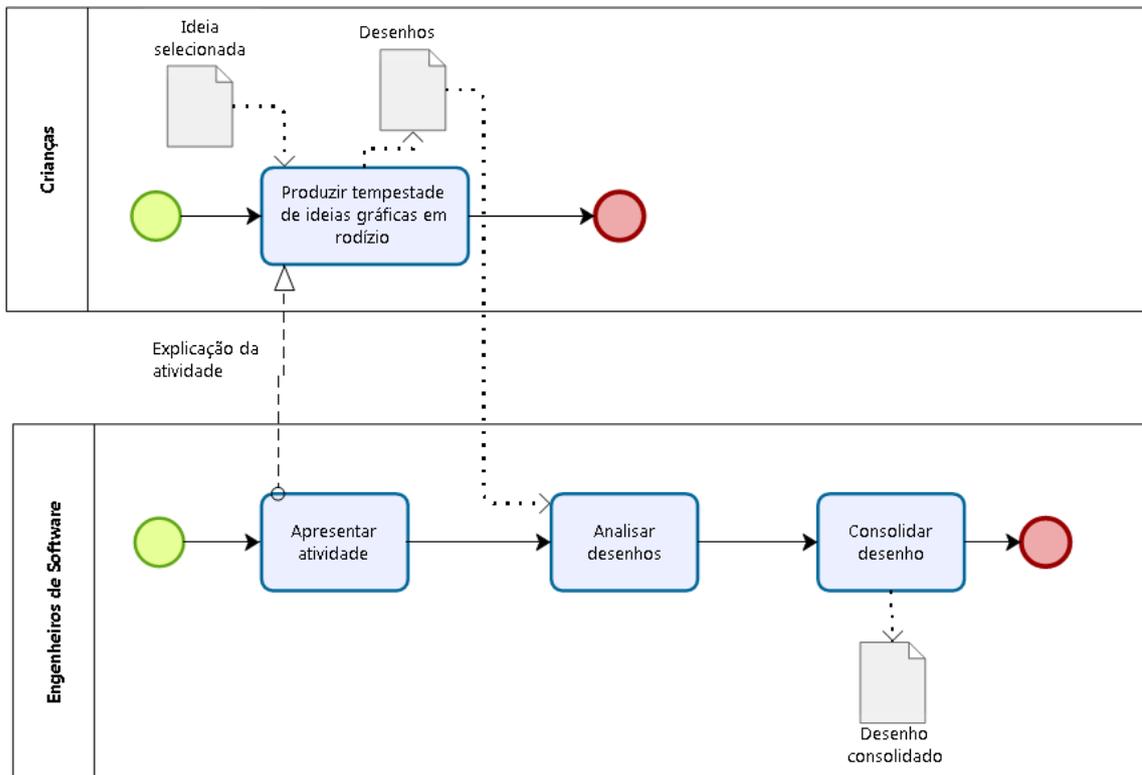
Figura 8 – Modelo do subprocesso Brainstorming



Fonte: Autoria Própria

No Braindrawing (Figura 9) (MULLER et al., 1997), os engenheiros de software iniciam pela apresentação da atividade para as crianças. Então, a partir da ideia selecionada para o desenvolvimento de software, as crianças produzem, em papéis, uma tempestade de ideias gráficas em rodizio com o propósito de contribuir para o desenho inicial da interface de usuário, em baixa fidelidade. Desenhos são gerados pelas crianças e analisados pelos engenheiros de software, que produzem uma proposta consolidada.

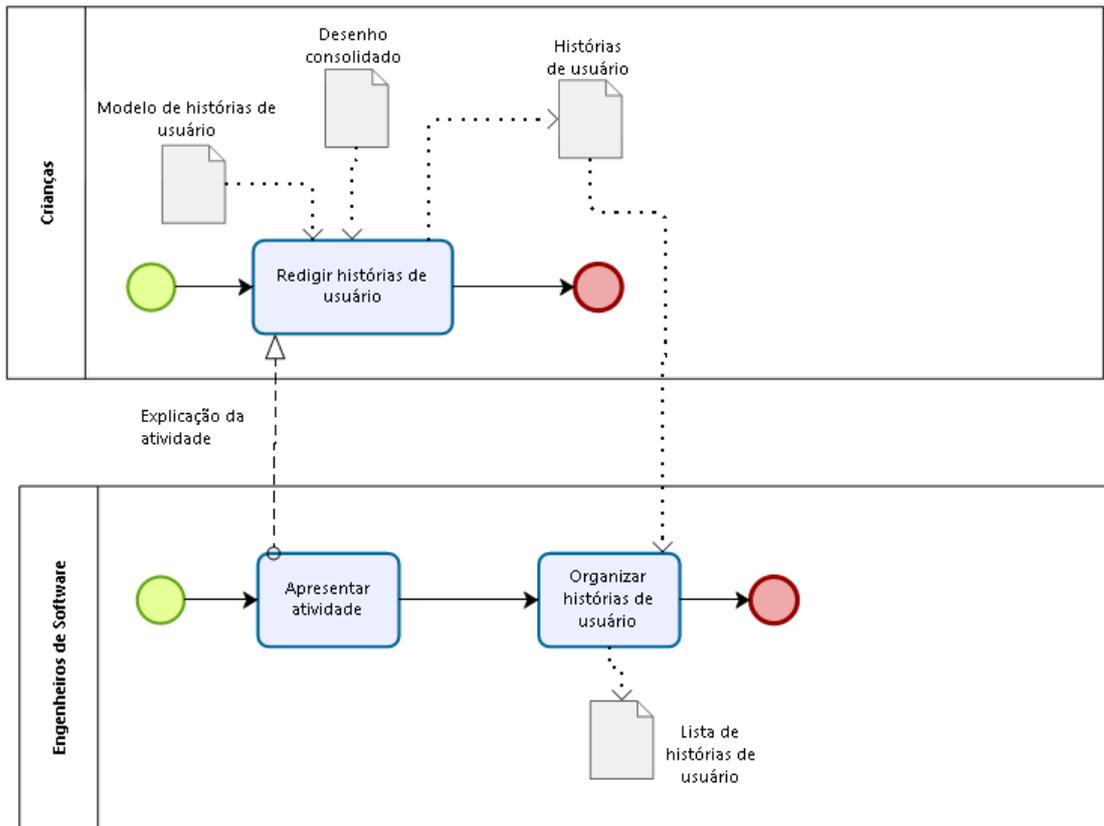
Figura 9 – Modelo do subprocesso Braindrawing



Fonte: Autoria Própria

Na atividade Levantar histórias de usuário (Figura 10) (COHN, 2014), os engenheiros de software apresentam a atividade para as crianças, que redigem uma série de histórias de usuário com o auxílio de um modelo de histórias de usuário, levando em conta o desenho consolidado do Braindrawing. A dinâmica dessa atividade pode ser adaptada, considerando a faixa etária das crianças envolvidas. Os engenheiros de software, então, finalizam a atividade organizando as histórias de usuário em uma única lista.

Figura 10 – Modelo do subprocesso Levantar histórias de usuário

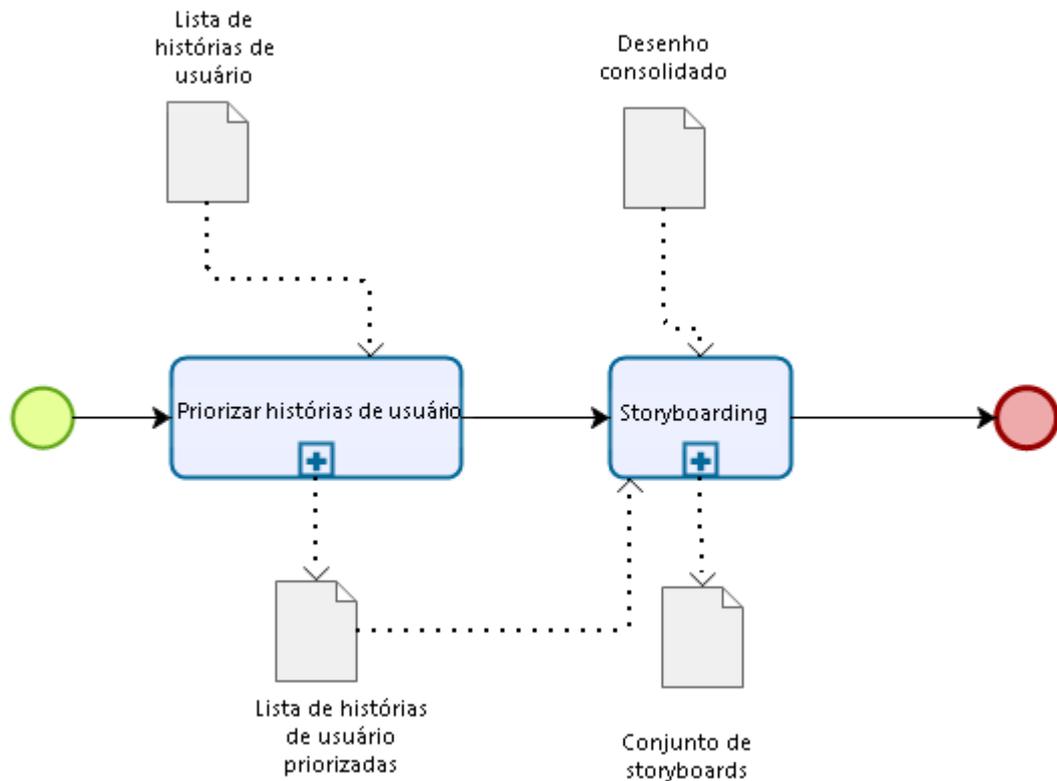


Fonte: Autoria Própria

#### 4.1.2 Analisar Requisitos

A Figura 11 descreve o subprocesso Analisar Requisitos, que organiza as atividades Priorizar histórias de usuário e Storyboarding. A atividade Priorizar histórias de usuário recebe como entrada uma lista de histórias de usuário e gera como saída uma lista de histórias de usuário priorizadas. Esta lista é entrada para a atividade Storyboarding, que também recebe como entrada o desenho consolidado do Braindrawing, produzindo como saída um conjunto de *storyboards*.

Figura 11 – Modelo do subprocesso Analisar Requisitos

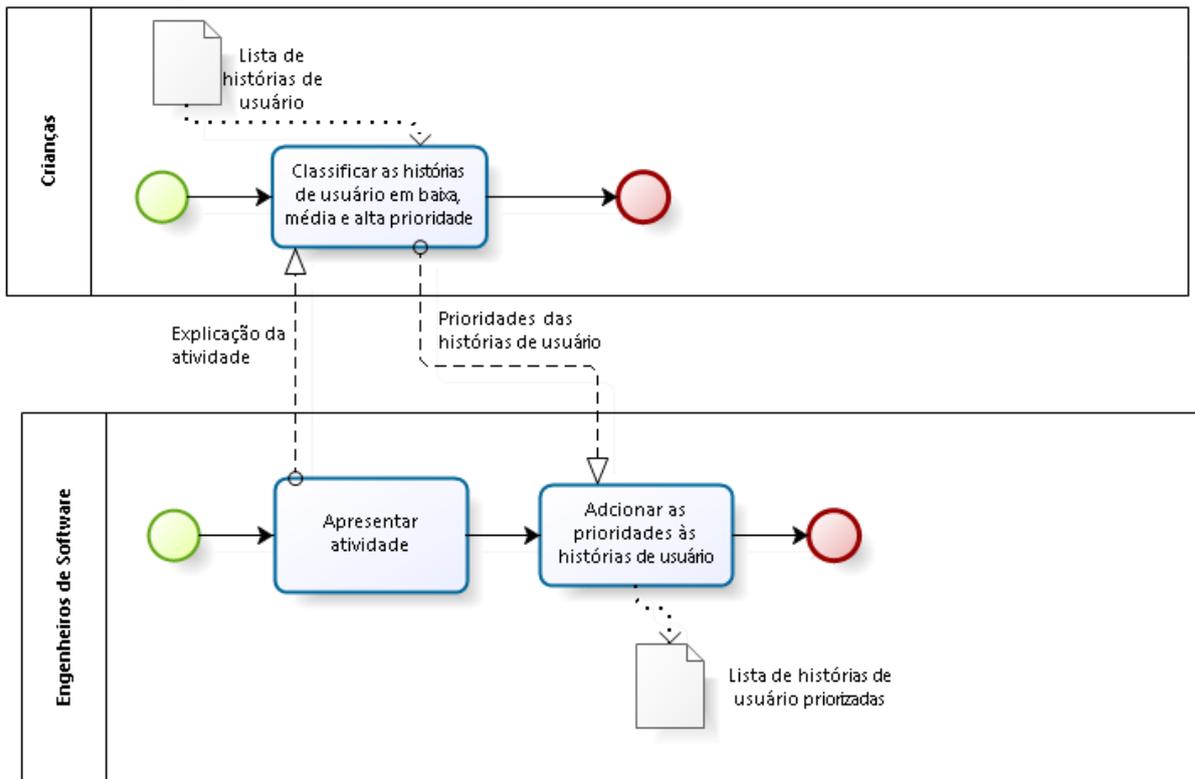


Fonte: Autoria Própria

A participação das crianças no subprocesso Analisar Requisitos é explicitada a partir do detalhamento de cada um dos subprocessos apresentados na Figura 11.

Na atividade Priorizar histórias de usuário (Figura 12), os engenheiros de software apresentam a atividade às crianças. Em seguida, as crianças, com a lista de histórias de usuário, classificam as histórias em baixa, média e alta prioridade. Por fim, os engenheiros de software adicionam as prioridades às respectivas histórias de usuário, gerando uma lista de histórias de usuários priorizadas como saída.

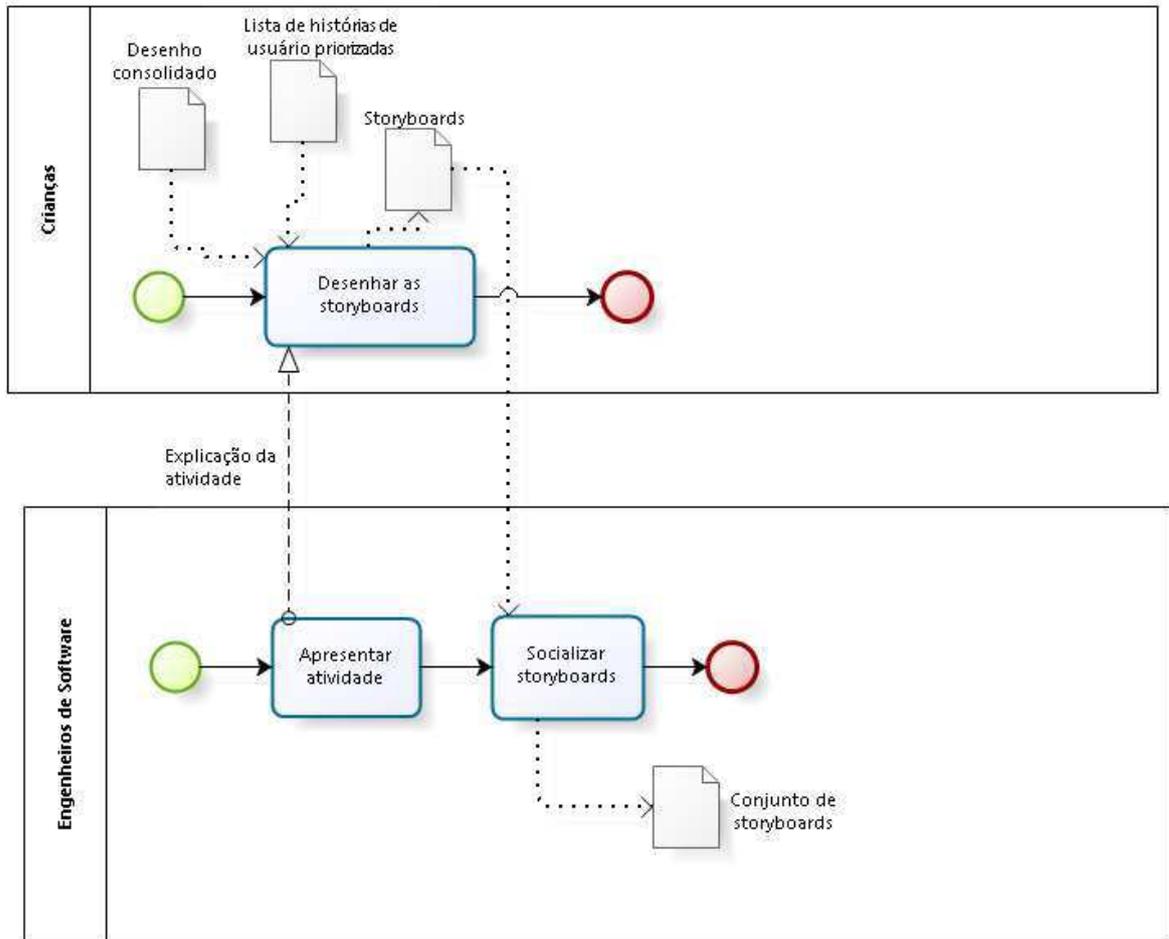
Figura 12 – Modelo do subprocesso Priorizar histórias de usuário



Fonte: Autoria Própria

Na atividade Storyboarding (Figura 13) (MEDEIROS *et al.*, 2004), os engenheiros de software apresentam a atividade para as crianças. Na sequência, usando como referência o desenho consolidado da atividade Braindrawing e a lista de histórias de usuário priorizadas as crianças desenham as *storyboards*, que representam a interação do usuário com o sistema. Com as storyboards em mãos, os engenheiros de software as socializam e as organizam em um conjunto de *storyboards*, que é saída da atividade.

Figura 13 – Modelo do subprocesso Storyboarding

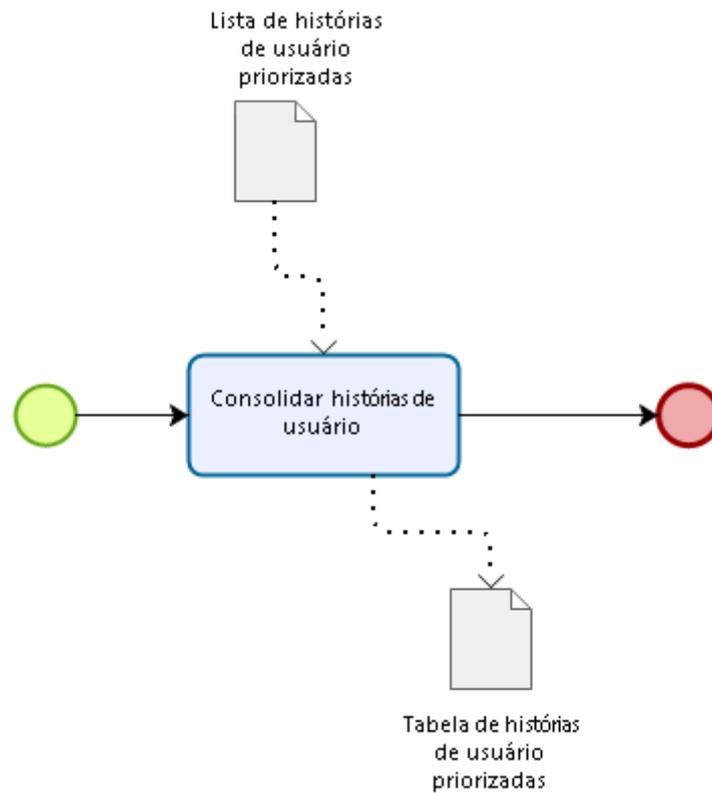


Fonte: Autoria Própria

### 4.1.3 Especificar Requisitos

A Figura 14 descreve o subprocesso Especificar Requisitos, que é realizado somente pelos engenheiros de software e possui somente a atividade Consolidar histórias de usuário. Esta atividade recebe como entrada uma lista de histórias de usuário priorizadas pelas crianças, que são revisadas e consolidadas, com o apoio de uma tabela. A revisão da lista de histórias de usuário inclui a correção de possíveis erros de português e de concordância verbal e a consolidação ocorre para organizar essa histórias de usuário em ordem de prioridade. Um exemplo de lista de histórias de usuário e de tabela de histórias de usuário pode ser encontrado no ANEXO C.

Figura 14 – Modelo do subprocesso Especificar Requisitos

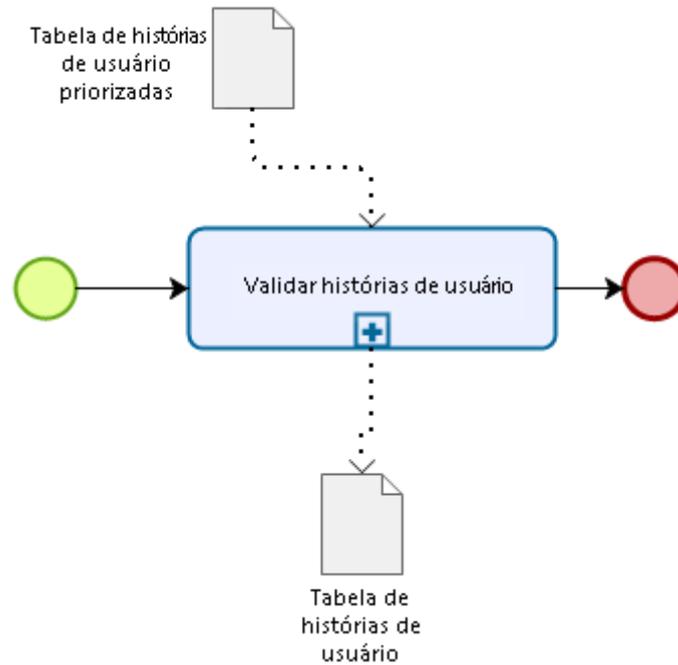


Fonte: Autoria Própria

#### 4.1.4 Validar Requisitos

A Figura 15 descreve o subprocesso Validar Requisitos, que contém somente a atividade Validar histórias de usuário. Esta possui como entrada uma tabela de histórias de usuário priorizadas e gera como saída uma tabela de histórias de usuário, que está priorizada e validada.

Figura 15 – Modelo do subprocesso Validar Requisitos

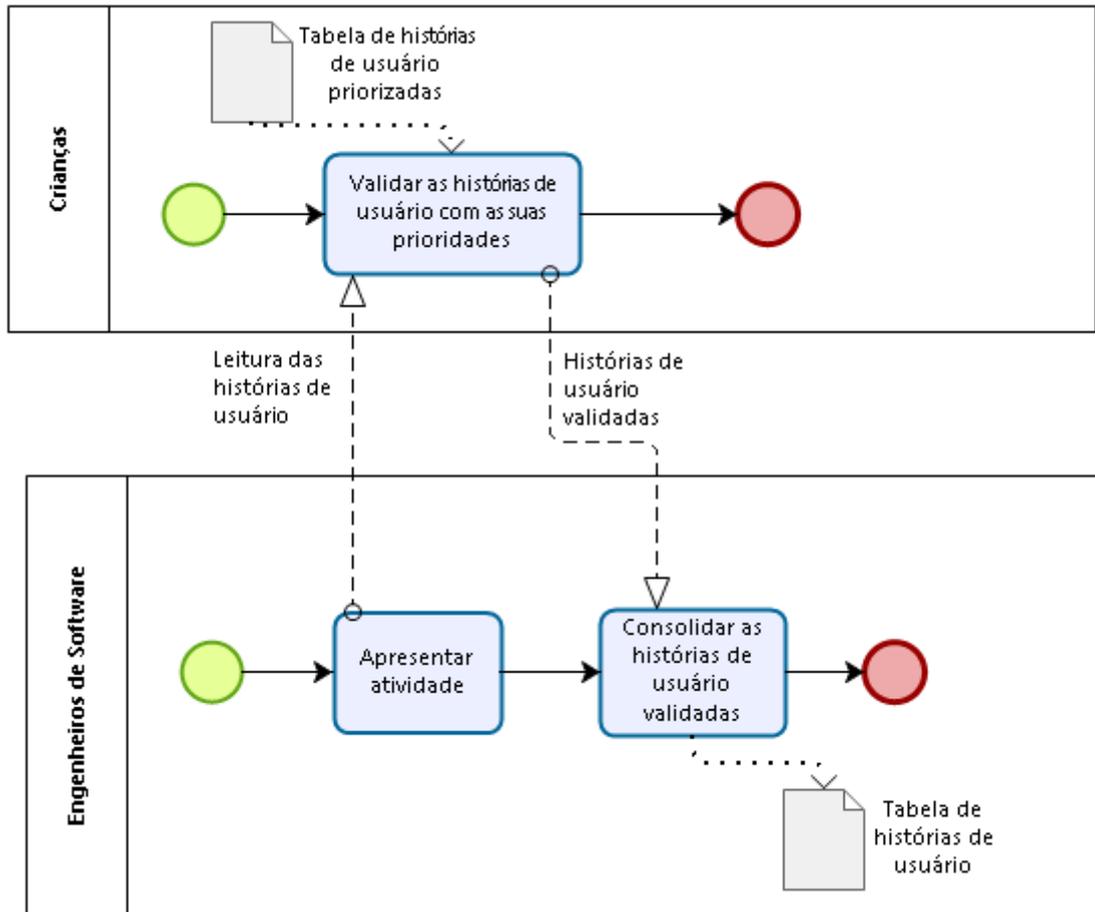


Fonte: Autoria Própria

A participação das crianças no subprocesso Validar Requisitos é explicitada a partir do detalhamento do subprocesso apresentado na Figura 15.

Na atividade Validar Histórias de Usuário (Figura 16), os engenheiros de software iniciam pela apresentação da atividade. As crianças prosseguem com a atividade, validando as histórias e suas prioridades recebidas como entrada em uma Tabela de histórias de usuário priorizadas. Nesse processo, as crianças indicam sua concordância ou discordância com as histórias organizadas pelos engenheiros de software. Os engenheiros de software, por fim, organizam as histórias de usuário validadas em uma tabela de histórias de usuário, que estão priorizadas e validadas.

Figura 16 – Modelo do subprocesso Validar histórias de usuário



Fonte: Autoria Própria

## 4.2 Um modelo de processo de desenvolvimento de software com a participação de crianças

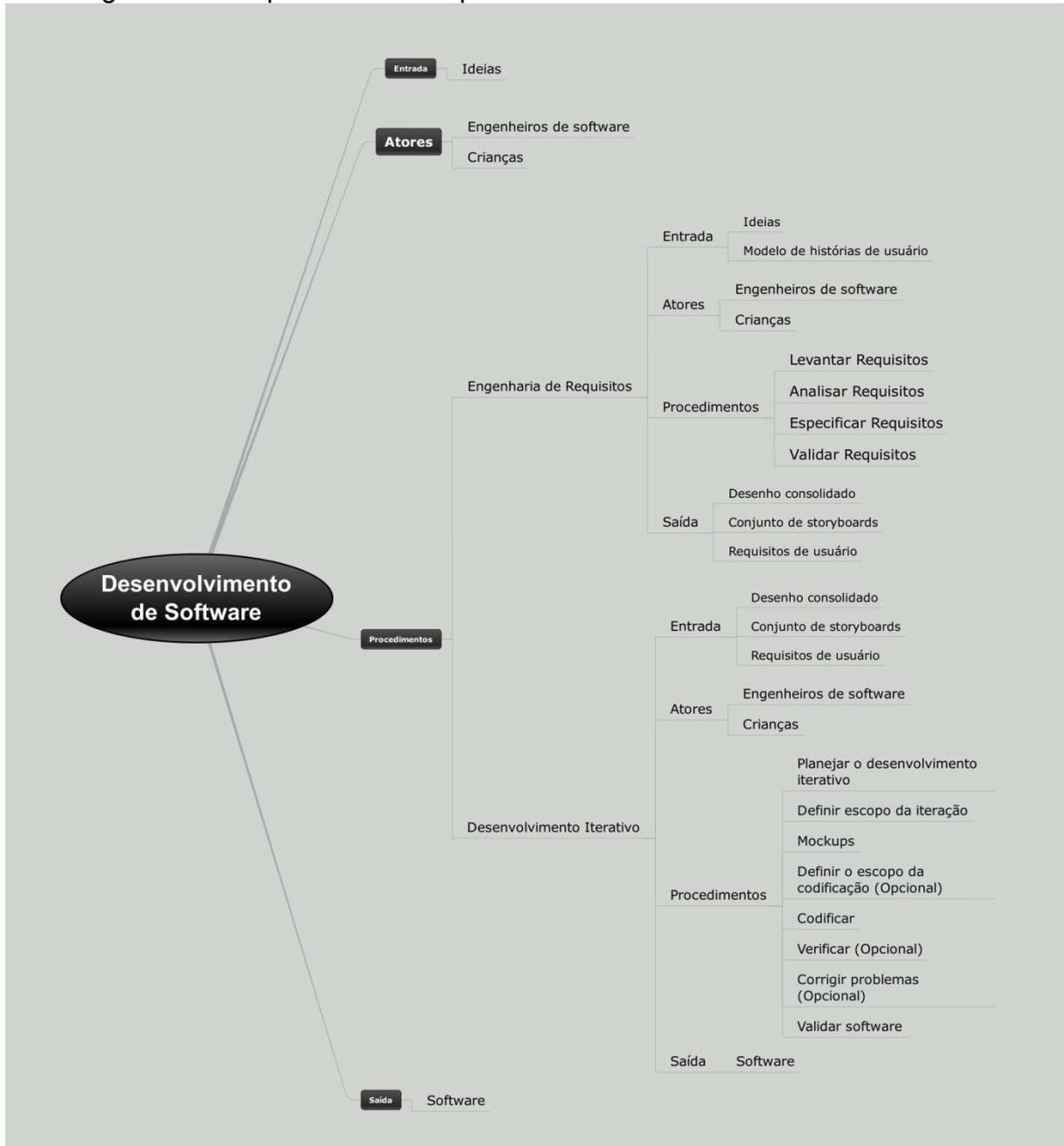
Outro resultado deste TCC é o modelo de processo de Desenvolvimento de Software (DS) para/com crianças, construído a partir do processo ER, da análise documental de outros dois processos de desenvolvimento de software para/com crianças no escopo do projeto de pesquisa DICA-TDIC, além de reuniões, no período de 08/08/2016 a 01/11/2016, com integrantes do projeto de pesquisa e com um docente com notório saber em modelagem de processo.

O mapa mental<sup>6</sup> (Figura 17) descreve as entradas, atores, procedimentos e saídas do processo. Cada novo ramo no mapa descreve níveis de abstrações mais baixos do processo. No mais alto nível de abstração, a entrada é uma ideia, os

<sup>6</sup> Mapa mental completo disponível em: <<https://www.mindomo.com/mindmap/mapa-mental-do-processo-de-desenvolvimento-de-software-atualizado-81431045931d429aa2edf9e31e01b99c>> Acesso em: 13/11/2016

atores são os engenheiros de software e as crianças, os procedimentos são Engenharia de Requisitos e Desenvolvimento Iterativo. Como saída, tem-se o software.

Figura 17 – Mapa mental de o processo de Desenvolvimento de Software



Fonte: Autoria Própria

Após a modelagem do mapa mental, os procedimentos foram organizados em ordem de execução, conforme (Quadro 4). Este quadro apresenta o fluxo dos procedimentos enumerados em ordem de acontecimento, com o setor descrevendo

os atores que são os engenheiros de software e as crianças, também evidenciando a participação das crianças em todo o processo.

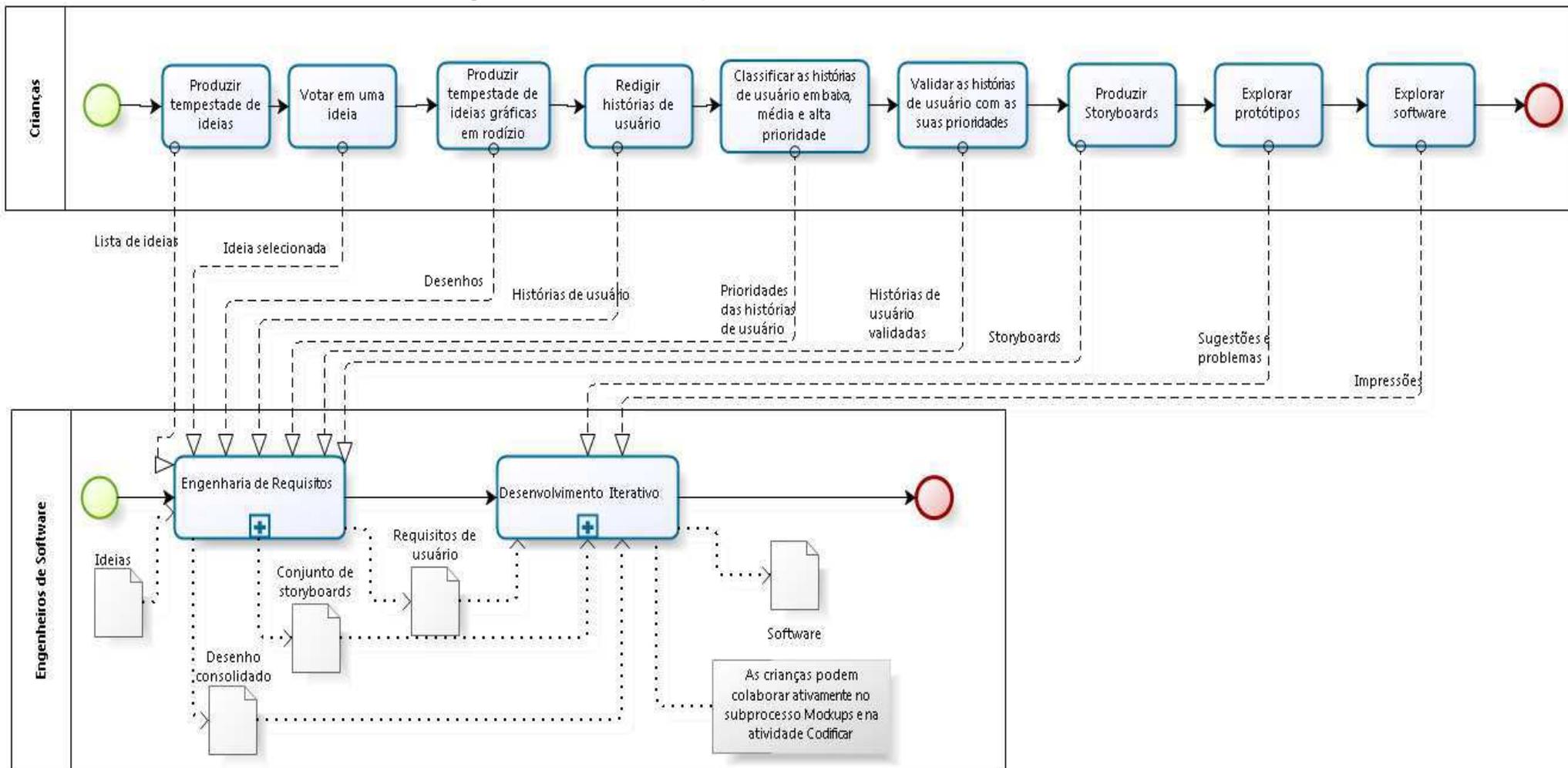
Quadro 4 – Fluxo de o processo Desenvolvimento de Software

<b>Fluxo de o processo Desenvolvimento de Software</b>		
<b>Passo</b>	<b>Setor</b>	<b>Procedimento</b>
1	Engenheiros de Software	Engenharia de Requisitos
	Crianças	
2	Engenheiros de Software	Desenvolvimento Iterativo
	Crianças	

Fonte: Autoria Própria

A Figura 18 representa o nível de abstração mais alto do processo DS, organizando os procedimentos documentados no Quadro 4. O processo DS apresenta os subprocessos Engenharia de Requisitos e Desenvolvimento Iterativo. O modelo, em notação BPMN, explicita as atividades realizadas pelas crianças. Essas atividades estão contidas nos subprocessos conduzidos pelos engenheiros de Software. Na troca das mensagens estão suas produções.

Figura 18 – Modelo do processo Desenvolvimento de Software

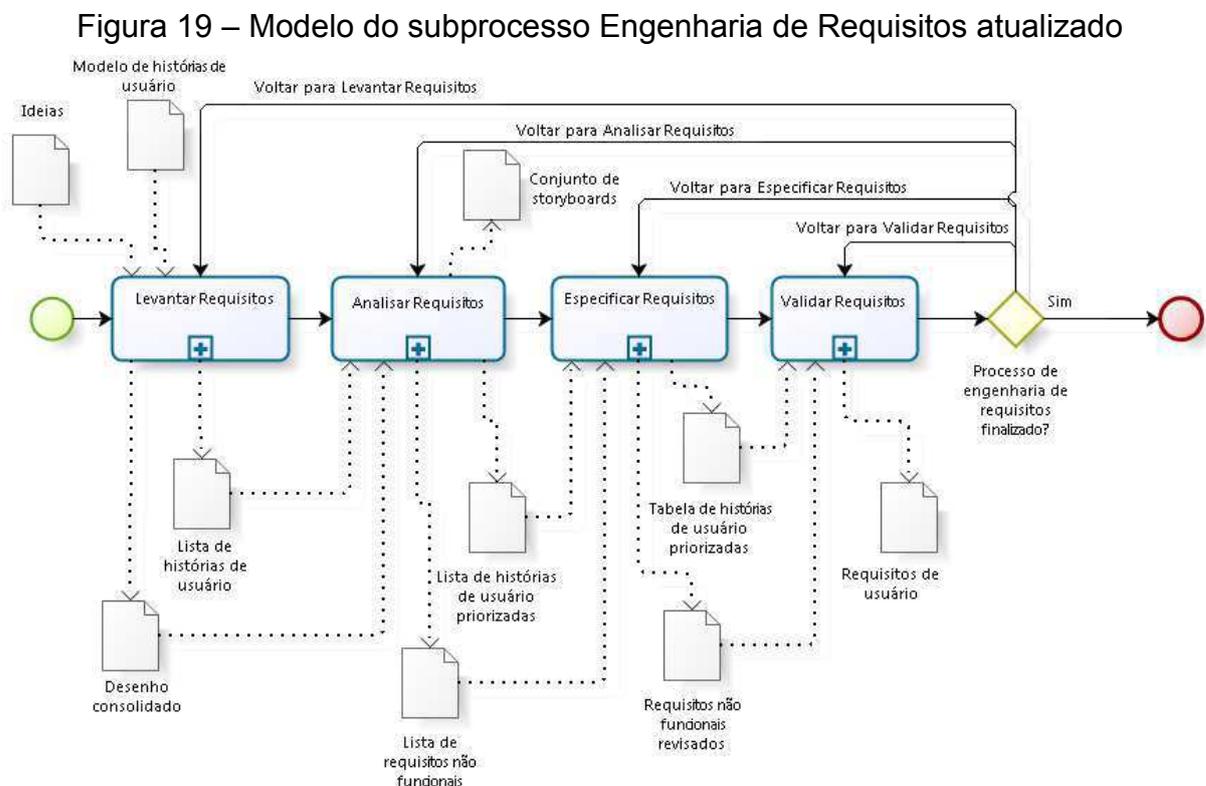


Fonte: Autoria Própria

Cada subprocesso do processo DS é descrito em detalhes nas próximas subseções.

#### 4.2.1 Engenharia de Requisitos

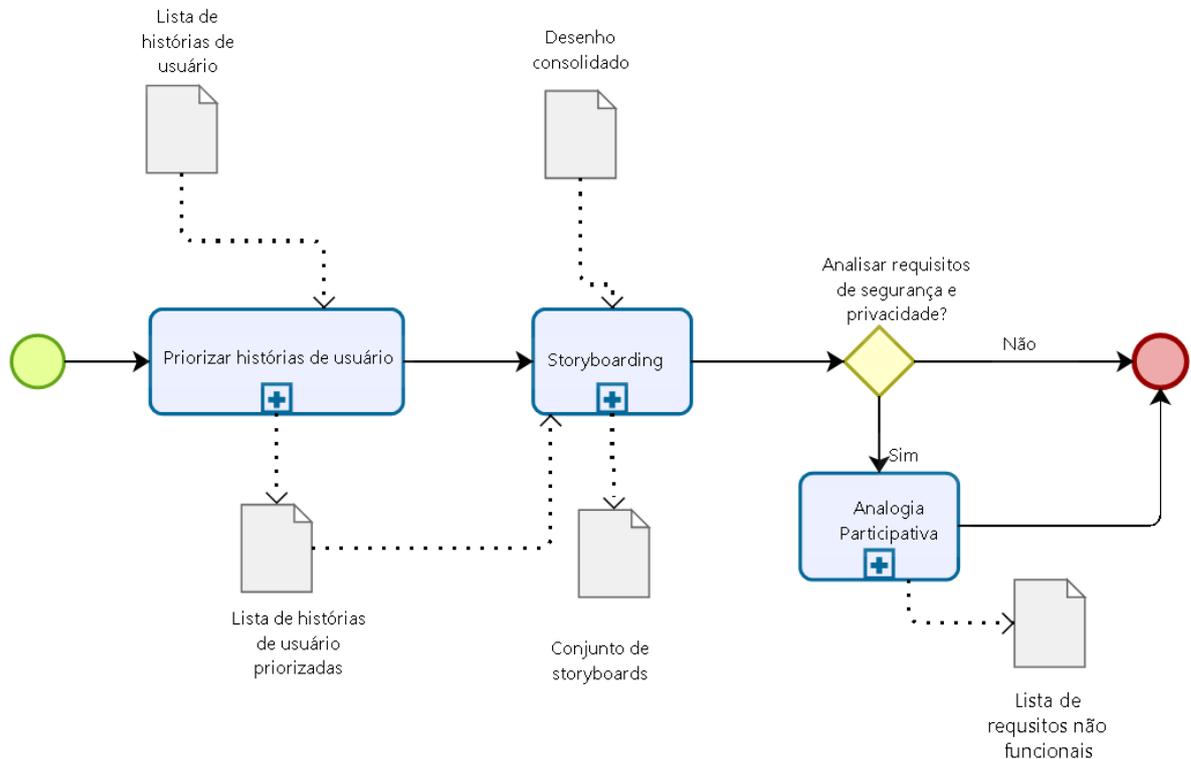
O subprocesso Engenharia de Requisitos está descrito na Seção 4.1. A Figura 19 apresenta algumas atualizações no processo.



Fonte: Autoria Própria

Ao subprocesso Analisar Requisitos, foi adicionada a saída lista de requisitos não funcionais. Alguns desses requisitos vêm da atividade opcional, Analogia Participativa (SOUZA, 2016), adicionada ao subprocesso Analisar Requisitos, conforme ilustrado na Figura 20. A lista de requisitos não funcionais é entrada para o subprocesso Especificar Requisitos, que também passa a gerar como saída uma lista de requisitos não funcionais revisada. Juntamente com a tabela de histórias de usuário priorizadas, são validados com as crianças, gerando o conjunto de requisitos de usuário.

Figura 20 – Modelo do subprocesso Analisar Requisitos



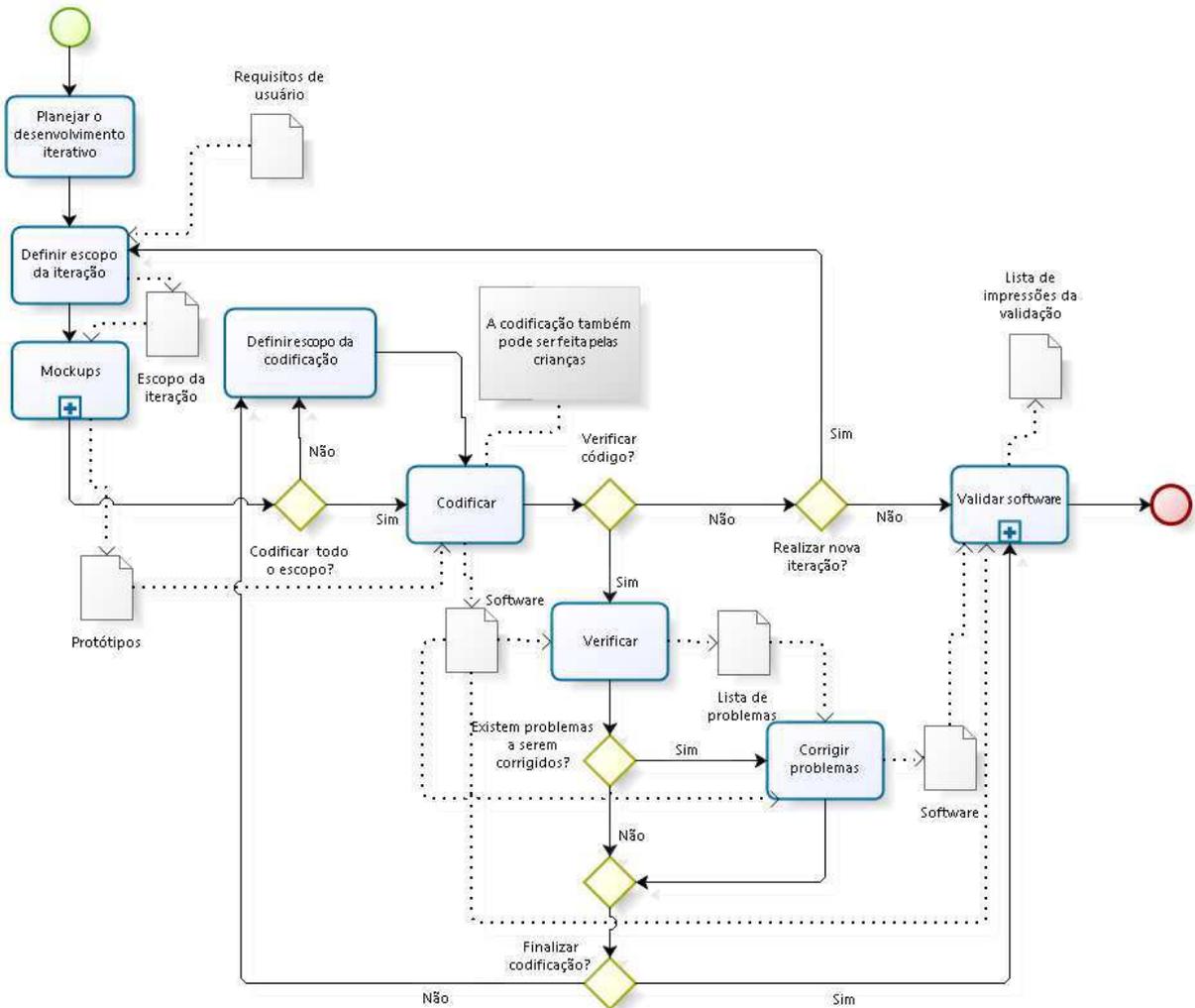
Fonte: Autoria Própria

A Analogia Participativa (APÊNDICE I), baseada em Souza (2016), serve para identificar requisitos de privacidade e de segurança utilizando uma analogia com as crianças. Por exemplo, adota-se um objeto que elas conhecem em analogia a um requisito e, verificando onde elas guardariam o objeto, compreende-se o grau de privacidade e de segurança daquele requisito. A atividade gera como saída uma lista de requisitos não funcionais.

#### 4.2.2 Desenvolvimento Iterativo

A Figura 21 descreve as atividades do subprocesso Desenvolvimento Iterativo (DI), baseado em outros dois processos de desenvolvimento de software propostos no contexto do projeto de pesquisa DICA-TDIC. Para chegar nesta proposta, além da análise documental, realizaram-se reuniões com integrantes do projeto de pesquisa, visando ao seu aprimoramento e validação.

Figura 21 – Modelo do subprocesso Desenvolvimento Iterativo



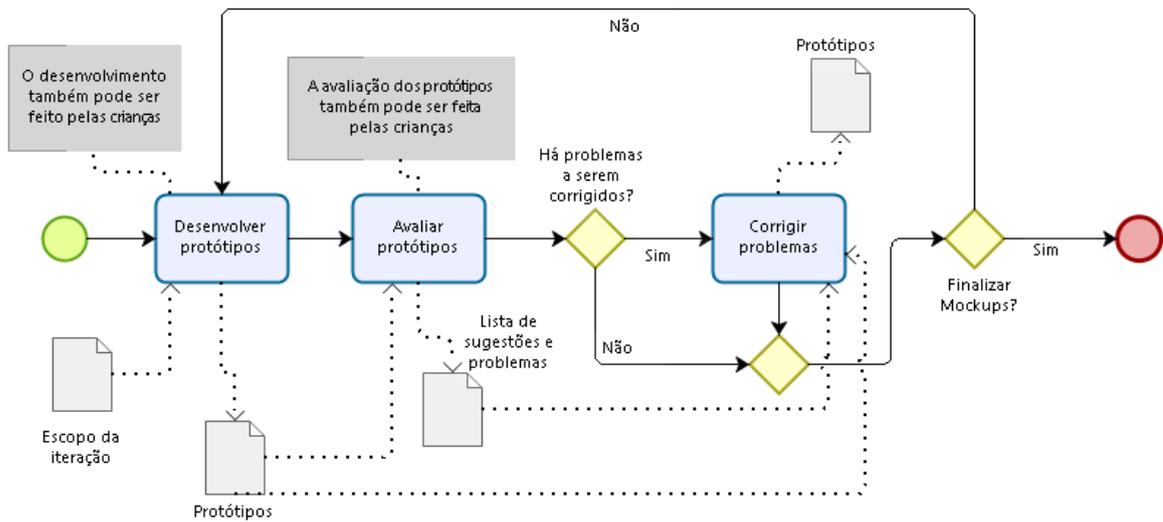
Fonte: Autoria Própria

Cada subprocesso do DI é descrito em detalhes a seguir.

O subprocesso é iniciado com a atividade Planejamento do desenvolvimento iterativo, onde ocorrem decisões como, por exemplo, se o desenvolvimento será iterativo e/ou incremental, os métodos e as técnicas de prototipação e de avaliação de interfaces de usuário a serem adotados, entre outras decisões relativas ao processo de desenvolvimento.

Na atividade Definir escopo da iteração, que recebe os requisitos de usuário como entrada, define-se como parte do escopo todos os requisitos ou parte deles. Neste último caso, assume-se que o processo como um todo é iterativo. Sobre o escopo definido, no subprocesso Mockups (MULLER et al., 1997) (Figura 22), desenvolvem-se, com a participação das crianças, protótipos da interface de usuário.

Figura 22 – Modelo do subprocesso Mockups



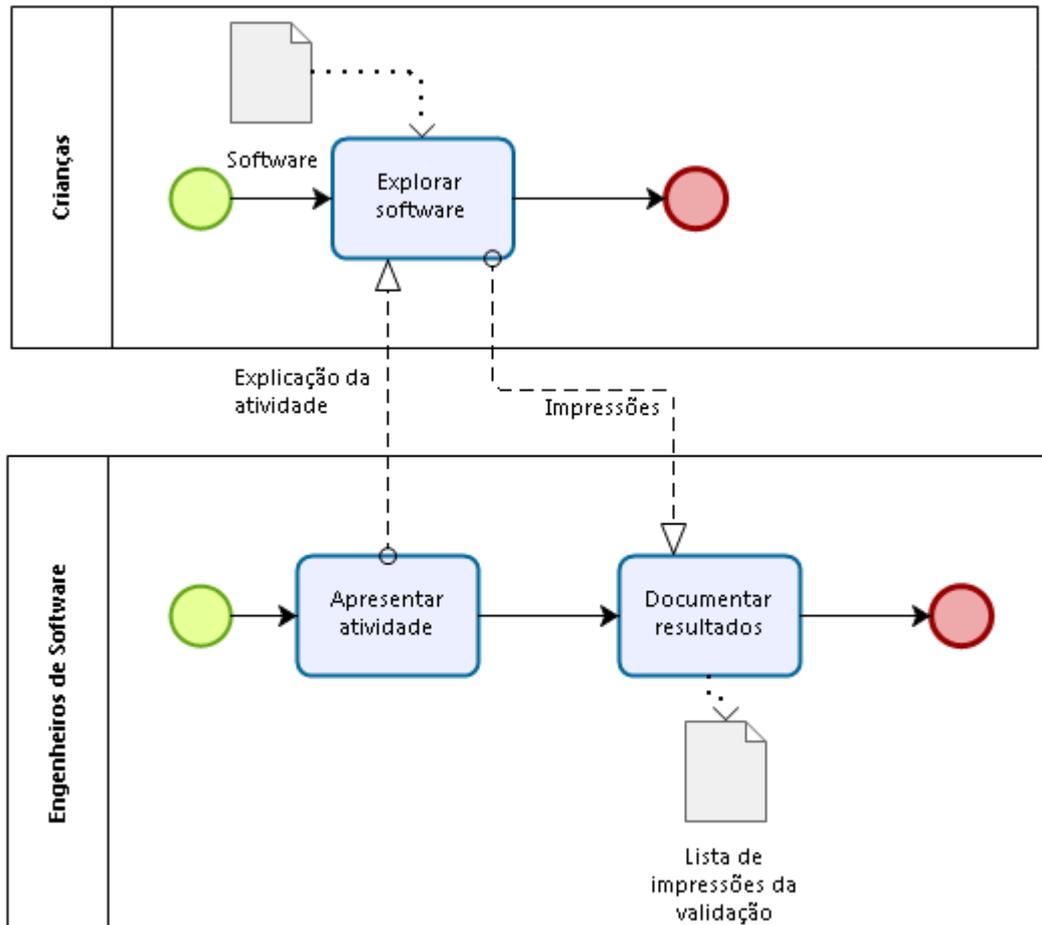
Fonte: Autoria Própria

O subprocesso Mockups organiza as atividades Desenvolver protótipos, Avaliar protótipos e Corrigir problemas. A primeira atividade, Desenvolver protótipos, pode ser realizada pelos engenheiros de software e/ou pelas crianças, gerando protótipos como saída. Esses protótipos são entradas para a atividade Avaliar protótipos, que gera como saída uma lista de sugestões e problemas. Essa avaliação também pode ser realizada pelos engenheiros de software e/ou pelas crianças, através de processos de inspeção, e/ou pelas crianças, de forma exploratória ou com o apoio de algum protocolo de avaliação com o usuário final. Então, ocorre um evento de decisão, que verifica se existem problemas a serem corrigidos. Caso existam, com os protótipos como entrada segue-se para a atividade Corrigir problemas. Após essa atividade que gera os protótipos como saída, o subprocesso Mockups pode ser finalizado ou continuar iterativamente.

Ao finalizar o subprocesso Mockups, antes da atividade Codificar, ocorre um evento de decisão para definir se todo o escopo será codificado como uma única atividade ou se a codificação será realizada de forma iterativa. A atividade Codificar gera software como saída. Essa atividade pode ser realizada pelas próprias crianças, por exemplo, com a adoção do ambiente Scratch (RIBEIRO, 2016). O software produzido serve de insumo para as atividades Verificar, Corrigir problemas e Validar software.

Uma vez que todo o software tenha sido completamente codificado, prossegue-se ao subprocesso Validar Software (Figura 23).

Figura 23 – Modelo do subprocesso Validar Software



Fonte: Autoria Própria

O subprocesso Validar Software organiza as atividades Apresentar atividade, Explorar software e Documentar resultados. Inicialmente, os engenheiros de software explicam a atividade. Após isso, as crianças exploram o software que receberam como entrada e relatam as impressões obtidas. Os engenheiros de software, então, documentam as impressões em uma lista que se torna saída do subprocesso.

Durante o processo DI, além de as crianças participarem do subprocesso Validar software, também podem participar ativamente do subprocesso Mockups e da atividade Codificar.

### **4.3 Considerações finais do capítulo**

Dentre as três maneiras de integração das áreas IHC e ES, apresentadas por Barbosa e Silva (2010), a opção escolhida para orientar o desenvolvimento deste trabalho foi a “Verificação de atividades de IHC que podem ser inseridas em processos de software”. Nesse sentido, a modelagem em notação BPMN permitiu expressar diferentes níveis de abstração de um processo de Engenharia de Software, assim como a integração de atividades de Interação Humano-Computador nesse processo.

Foi um desafio modelar a colaboração das crianças, evidenciada pelas práticas de IHC. Esse desafio foi superado pela identificação e avaliação de alternativas de representação da notação BPMN. Outros desafios envolvidos no desenvolvimento deste trabalho incluem a manutenção da consistência entre modelos (ex.: Mapa Mental e modelo em notação BPMN); o próprio processo de abstração de diferentes experiências em uma única representação; e as tomadas de decisão, através de processos de negociação, sobre uso de termos para denominar processos, subprocessos, atividades e seus responsáveis, documentos etc.

## 5 CONSIDERAÇÕES FINAIS

Crianças, ao serem incluídas no desenvolvimento de tecnologias para o seu uso, oferecem contribuições importantes. Um processo de desenvolvimento de software, que inclua crianças ativamente, dá oportunidade para que informem o que, como e porque querem.

Para desenvolver o modelo do processo de desenvolvimento de software para/com crianças, documentado nesta monografia, foi necessário rever e aprofundar conhecimentos desenvolvidos durante a graduação, como estratégias de integração ES e IHC, modelos de processos, modelagem de processos, entre outros. A produção do modelo de processo também contou com a análise sistematizada de experiências e de registros das atividades e práticas de quatro estudos exploratórios de desenvolvimento de software que aconteceram no contexto do projeto de pesquisa DICA-TDIC.

O modelo de processo proposto, em notação BPMN, provê apoio a engenheiros de software e pesquisadores para desenvolverem software para/com crianças, que contemplem mais adequadamente a participação, os interesses, os gostos, as necessidades e as experiências desses usuários em uma perspectiva de ES. O modelo integra atividades de IHC a processos de Engenharia de Software, explicitando possibilidades de colaboração de crianças.

Destacam-se como limitações do trabalho a documentação do levantamento dos requisitos não funcionais e do subprocesso Evolução de Software. Num primeiro momento, as experiências de Engenharia de Requisitos com as crianças não contemplaram explicitamente o levantamento de requisitos não funcionais e, como resultado da análise documental, deu-se ênfase à análise de requisitos não funcionais de segurança e privacidade. Já a Evolução de Software envolve acompanhar o software após sua entrega. Essa experiência, entretanto, não ocorreu no escopo do projeto de pesquisa DICA-TDIC acarretando da não definição desse subprocesso no processo proposto.

Entre os trabalhos futuros, o modelo poderá subsidiar o desenvolvimento de software com crianças em novos projetos. Além disso, poderá ser revisado e aprimorado, por exemplo, contemplando o subprocesso Evolução, que envolve acompanhar um software após sua entrega.

## REFERÊNCIAS

- ALMEIDA, P. A. **Curso de Mapeamento de Processos de Trabalho com BPMN e Bizagi**. Disponível em: <<http://portal.tcu.gov.br/lumis/portal/file/fileDownload.jsp?inline=1&fileId=8A8182A24F0A728E014F0B27E6D51C6C>>. Acesso em: 15 abr. 2016.
- APPOLINÁRIO, Fabio. **Dicionário de metodologia científica: um guia para a produção do conhecimento científico**. São Paulo: Atlas, 2004.
- BARBOSA, S. D. J; SILVA, B.S. **Interação Humano-Computador**. Rio de Janeiro: Elsevier, 2010.
- BORGES, Luciana Correa Lima de Faria *et al.* **Customizing a communication device for a child with cerebral palsy using participatory design practices: contributions towards the PD4CAT method**. In: BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, 11., 2012, Cuiabá. Anais.... Porto Alegre: SBC, 2012. Disponível em: <<http://dl.acm.org/citation.cfm?id=2393544>>. Acesso em: 07 abr. 2016.
- BOSCARIOLI, Clodis *et al.* **Avaliação e design de interação de jogos voltados ao aprendizado de crianças surdas**. In: BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS, 11. 2012, Cuiabá. Anais... Porto Alegre: SBC, 2012. Disponível em: <<http://dl.acm.org/citation.cfm?id=2400086>>. Acesso em: 14 abr. 2016.
- BRAZ, Priscilla; RAPOSO, Alberto; SOUZA, Clarisse Sieckenius. **Uso de design probes no design de tecnologias para terapeutas de crianças com autismo**. In: SIMPÓSIO BRASILEIRO SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 13. 2014, Foz do Iguaçu. Anais... Porto Alegre: SBC, 2014. Disponível em: <<http://dl.acm.org/citation.cfm?id=2738079>>. Acesso em: 14 abr. 2016.
- BUZAN, T. **Mapas mentais e sua elaboração: um sistema definitivo de pensamento que transformará a sua vida**. São Paulo: Cultrix, 2005.
- CARTAXO, Bruno *et al.* **The Impact of Scrum on Customer Satisfaction: an Empirical Study**. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 27. 2013, Brasília. Anais... Porto Alegre: SBC, 2013. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6800189>>. Acesso em: 01 abr. 2016.
- CARVALHO, A. M. B. R.; CHIOSSI, T. C. d. S. C. **Introdução à Engenharia de Software**. [S. I.]: Editora Unicamp, 2001. ISBN 9788526805019.
- COHN, M. **User stories applied: for agile software development**. [S. I.]: Addison-Wesley Professional, 2004.
- FERNANDES, Priscila; CONTE, Tayana; BONIFÁCIO, Bruno. **WE-QT: A Web Usability Inspection Technique to Support Novice Inspectors**. In: SIMPÓSIO

BRASILEIRO DE ENGENHARIA DE SOFTWARE, 26. 2012, Natal. Anais... Porto Alegre: SBC, 2012. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6337889>>. Acesso em: 01 abr. 2016.

FERREIRA, Bruna; CONTE, Tayana; BARBOSA, Simone Diniz Junqueira. **Eliciting Requirements using Personas and Empathy Map to Enhance the User Experience**. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 29. 2015, Belo Horizonte. Anais... Porto Alegre: SBC, 2015. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=7328012&tag=1](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7328012&tag=1)>. Acesso em: 01 abr. 2016.

FERREIRA, M. A. M. **Design inclusivo e participativo na Web**: incluindo pessoas surdas, mestrado em ciência da computação. 2014.

GRAY, Stuart; ROBERTSON, Judy; RAJENDRAN, Gnanathusharan. **BrainQuest**: an active smart phone game to enhance executive function. In: INTERACTION DESIGN AND CHILDREN, 14. 2015, Medford. Anais... New York: ACM Digital Library, 2015. Disponível em: <<http://dl.acm.org/citation.cfm?id=2771846>> Acesso em: 07 abr. 2016.

HANSEN, Elin Irene Krog; IVERSEN, Ole Sejer. **You are the real experts: Studying teenagers' motivation in participatory design**. In: INTERACTION DESIGN AND CHILDREN, 12. 2013, New York: Anais eletrônicos. New York: ACM Digital Library, 2013. Disponível em: <<http://dl.acm.org/citation.cfm?id=2485826>> Acesso em: 19 mai. 2016.

HEWETT, T. Thomas. **ACM SIGCHI Curricula for Human-Computer Interaction**. Nova York, 1992. ACM Digital Library. Disponível em: <<http://dl.acm.org/citation.cfm?id=2594128&CFID=694297502&CFTOKEN=39986011>> Acesso em: 16 nov. 2016.

LARROSSA, Luciano. **O que é um Mapa Mental e como utilizar para Melhorar a sua Produtividade**. Disponível em: <<http://www.escolafreelancer.com/o-que-e-um-mapa-mental-e-como-utilizar-para-melhorar-a-produtividade/>>. Acesso em: 15 abr. 2013.

LEITE, Jair; BATISTA, Thais; LEITE, Larissa. **Software Engineering Research in Brazil**: an analysis of the last five editions of SBES. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 25. 2011, São Paulo. Anais eletrônicos. Porto Alegre: SBC, 2011. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6065142>>. Acesso em: 01 abr. 2016.

LÔBO, Kézia Letícia da Silva. **Uma abordagem ética e responsável ao design para/com crianças**: investigando a integração de práticas de interação humano-computador à engenharia de requisitos. 2016. 93 f. TCC (Graduação) - Curso de Engenharia de Software, Universidade Federal do Pampa- Unipampa, Alegrete, 2016. CD-ROM.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Metodologia Científica**. 7ª Edição. São Paulo: Atlas S.A., 2009.

MEDEIROS, L. *et al.* **Uso de storyboards para a documentação de requisitos no desenvolvimento distribuído de software**. 2004.

MULLER, M. J.; HASLWANTER, J. H.; DAYTON, T. **Participatory Practices in the Software Lifecycle**. Elsevier Science, 1997. ISBN 9780080632882. Disponível em: <<https://books.google.com.br/books?uid=117183610082700690402&hl=pt-BR>> Acesso em: 18 de out. 2016e.

PAZMINO, Priscilla F. Jimenez; SLATTERY, Brian; LYONS, Leilah. **Designing for youth interpreter professional development: a sociotechnologically-framed participatory design approach**. In: INTERACTION DESIGN AND CHILDREN, 14. 2015, Medford. Anais... New York: ACM Digital Library, 2015. Disponível em: <<http://dl.acm.org/citation.cfm?id=2771840>> Acesso em: 07 abr. 2016.

PREECE, J.; ROGERS, Y.; SHARP, H. **Design De Interação**. Porto Alegre; São Paulo: Elsevier, 2005.

PRESSMAN, R.S. **Engenharia de Software**. 6. ed. Rio de Janeiro: Mc Graw-Hill, 2006.

PUCCINI, Angela M.; PUCCINI, Marina; CHANG, Angela. **Acquiring educational access for neurodiverse learners through multisensory design principles**. In: INTERACTION DESIGN AND CHILDREN, 12. 2013, New York: Anais eletrônicos. New York: ACM Digital Library, 2013. Disponível em: <<http://dl.acm.org/citation.cfm?id=2485848>> Acesso em: 19 mai. 2016.

RIBEIRO, S. S.; MELO, A. M. **Integração IHC-ES no Desenvolvimento de Software para/com/por Crianças com Scratch**. In: SIMPÓSIO SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 15., 2016, São Paulo. Anais... SBC: Porto Alegre, 2016. (no prelo)

RIVERO, Luis; CONTE, Tayana. **Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications**. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 26. 2012, Natal. Anais... Porto Alegre: SBC, 2012. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6337861>>. Acesso em: 01 abr. 2016.

RIVERO, Luis; KAWAKAMI, Guto; CONTE, Tayana. **Using a Controlled Experiment to Evaluate Usability Inspection Technologies for Improving the Quality of Mobile Web Applications Earlier in their Design**. In SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 28. 2014, Maceió. Anais... Porto Alegre: SBC, 2014. Disponível em:<<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6943493>>. Acesso em: 01 abr. 2016.

SARMENTO, Tadeu. **Guia Geral de Processos de Negócios - Atendimento**. Disponível em: <<http://conhecimento.consensotec.com.br/doku.php?id=treinamentos:livre:processos:atendimento:processo-macro>>. Acesso em: 10 mar. 2016.

SEGUI, Bruno da Silva. **Desenvolvimento de sistema para solicitação de registro de marca de gado**. Alegrete: Universidade Federal do Pampa, 2015. 1 CD ROM.  
TOLFO, C. Metodologia de mapeamento e modelagem de processos. Alegrete: [s.n.], 2013. 20-55 p.

SOMMERVILLE, I. **Engenharia de Software**. 8ª ed. São Paulo. Pearson Education, 2007.

SOUZA, A. T.; MELO, A. M. **Análise de Requisitos de Segurança e Privacidade com Crianças através do Método de Analogia Participativa**. In: SALÃO INTERNACIONAL DE ENSINO, PESQUISA E EXTENSÃO, 8., 2016, Uruguaiana. Anais... UNIPAMPA: Bagé, 2016. (no prelo)

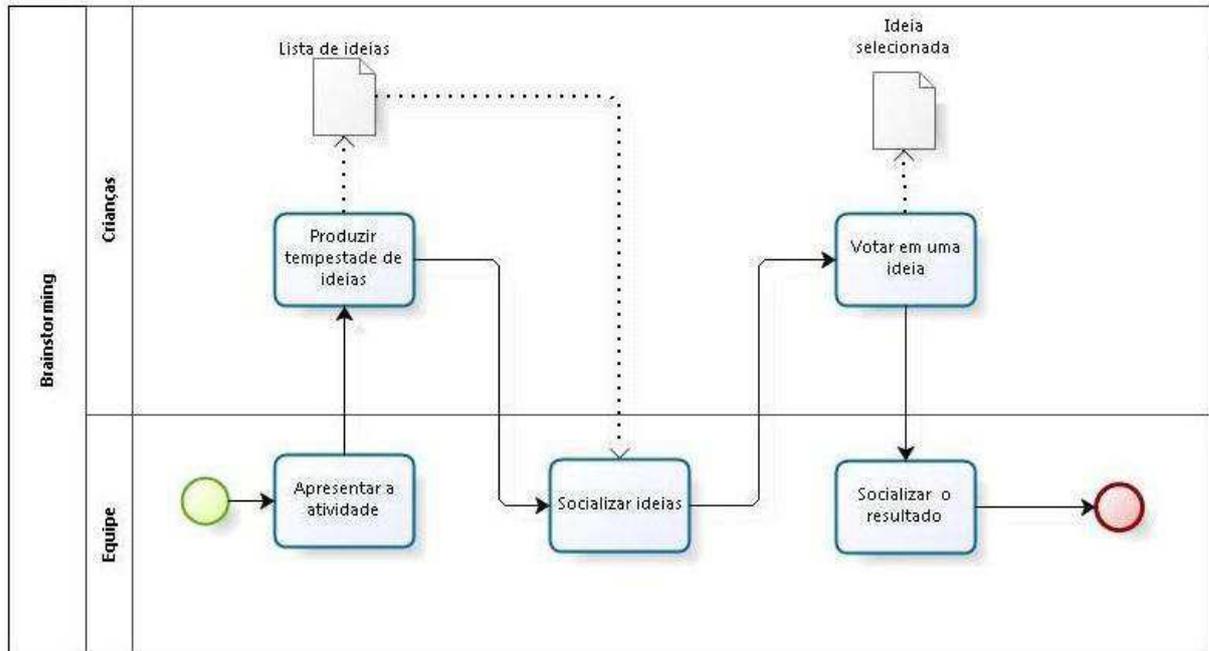
UNICEF, Fundo das Nações Unidas para a Criança. **Convenção sobre os Direitos da Criança**. Nova Iorque, 1989. Disponível em: <[http://www.unicef.org/brazil/pt/resources\\_10120.htm](http://www.unicef.org/brazil/pt/resources_10120.htm)>. Acesso em: 11 nov. 2016.

VALENTIM, Natasha M. Costa; CONTE, Tayana. **Improving a Usability Inspection Technique based on Quantitative and Qualitative Analysis**. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 28. 2014, Maceió. Anais eletrônicos. Porto Alegre: SBC, 2014. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6943494>>. Acesso em: 01 abr. 2016.

## APÊNDICES

### APÊNDICE A – Um processo com dois atores

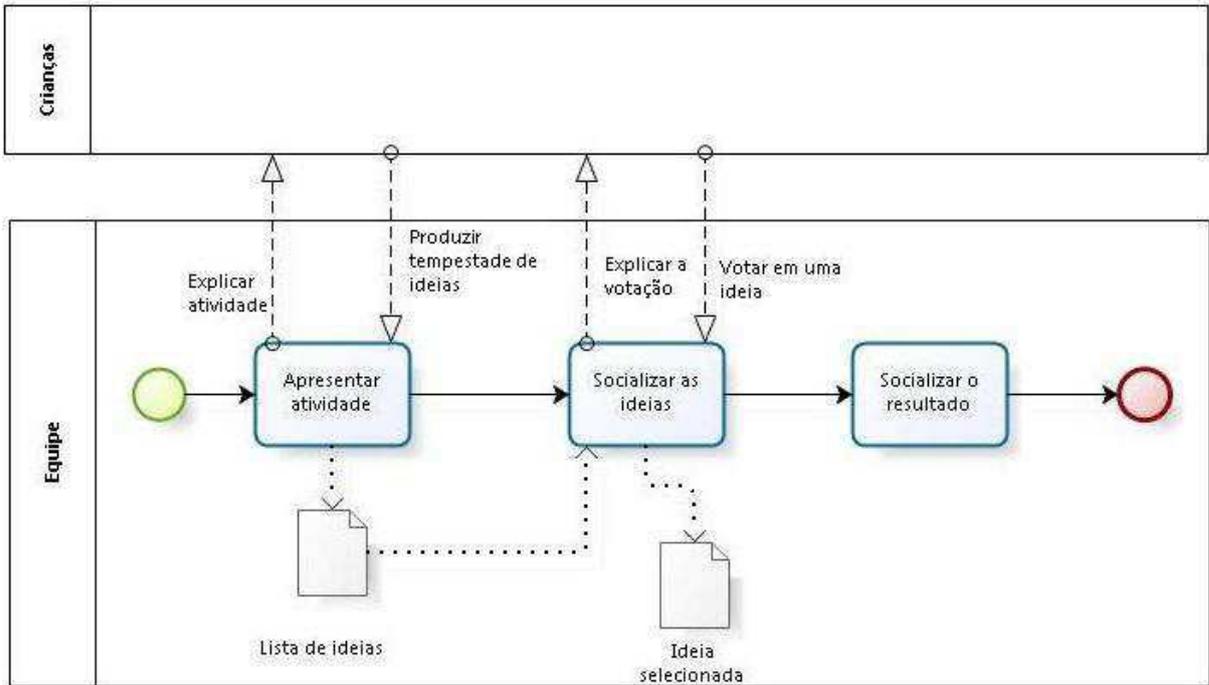
Figura 24 – Exemplo de processo com dois atores



Fonte: Autoria própria

## APÊNDICE B – Um processo para cada ator, sem descrever um dos processos

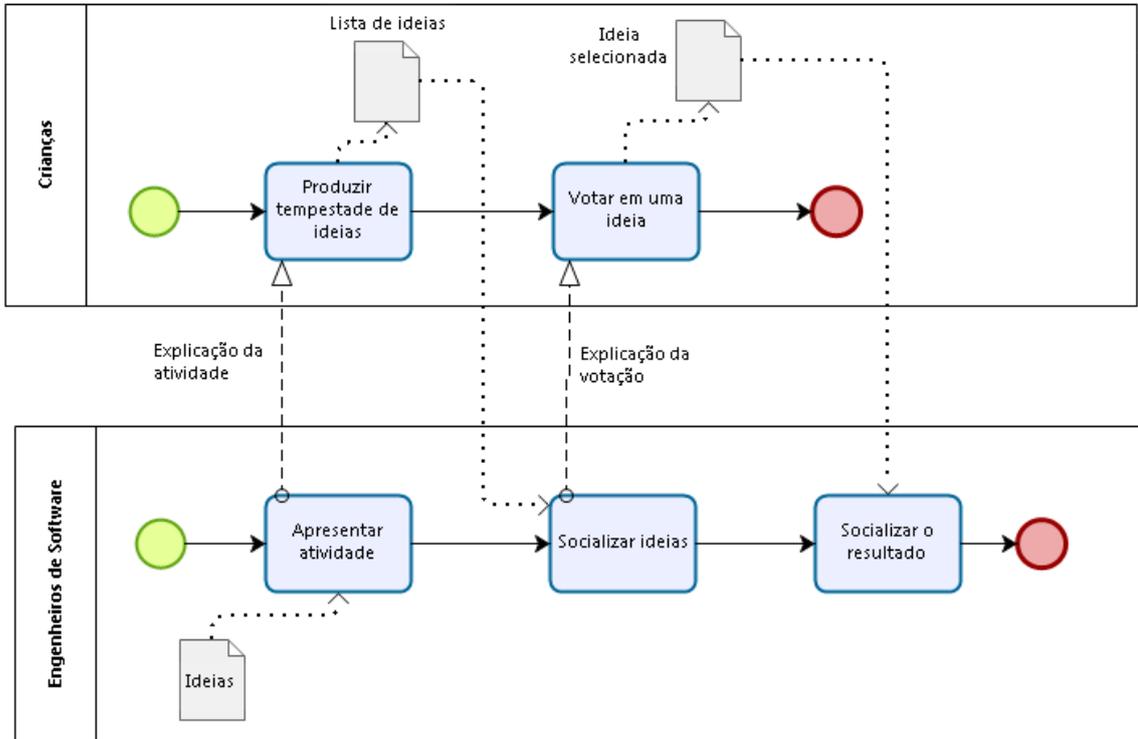
Figura 25 – Exemplo dum processo para cada ator



Fonte: Autoria própria

## APÊNDICE C – Um processo para cada ator, descrevendo ambos os processos

Figura 26 – Um processo para cada ator, ambos descritos



Fonte: Autoria própria

## APÊNDICE D – Roteiro de entrevista do processo 1

### Perguntas ao entrevistado 1.

1. Se os requisitos já estão especificados, de que modo a análise de requisitos de segurança e privacidade irá impactar nessa especificação antes da validação? Não estaria faltando uma iteração entre especificação e análise?

Resposta: Requisitos Funcionais do outro estudo, elicita e especifica os não funcionais e os de segurança e privacidade. Analisa os requisitos de privacidade através da analogia participativa a fim de analisar os dados sensíveis da interface e por fim valida- os. Sendo assim não é necessária a iteração.

2. Como será a verificação e validação dos requisitos de segurança e privacidade? Há algum método para apoiar esse subprocesso?

Resposta: Os requisitos em geral serão validados no seguinte método. Serão entregues às crianças os protótipos, as histórias de usuários e os requisitos não funcionais e serão aplicadas questões para verificar e validar se está tudo ok.

3. Como será a correção dos problemas? O que está incluído nessa atividade?

Resposta: Vai ser levado o software para as crianças testarem por fim de sprint e após será feita a correção de erros encontrados.

4. Os dias estimados para as atividades seriam os mesmos se o prazo fosse outro?

Resposta: Poderiam mudar, mas muito pouco. Na etapa, acredito que haveria uma mudança considerável.

## APÊNDICE E – Roteiro de entrevista do processo 2

Perguntas ao entrevistado 2.

1. Qual o tempo estimado para um ciclo do Design Iterativo?

Resposta: 1 mês e 1 semana, + ou - 11 encontros de 2 horas de duração cada.

2. Como será a validação do produto? Definiu-se um processo para sua realização?

Resposta: Presencial de forma oral. Serão feitas perguntas para verificar se o software está de acordo com as expectativas deles.

## APÊNDICE F – Roteiro de entrevista sobre os dois processos

Perguntas para os dois entrevistados:

1. Existe alguma atividade que vocês acreditam que possa ser omitida ou substituída por outra? Quais? Em caso, de substituição por quais?

Respostas:

Entrevistado 1: Storyboarding.

Entrevistado 2: Acredito que talvez a Storyboarding possa ser omitida.

2. Existe alguma atividade do processo que vocês acreditam que, a cada aplicação, necessite ser modificada ou adaptada a contextos diferentes?

Respostas:

Entrevistado 1: Analogia Participativa talvez deva ser modificada dependendo da idade da criança, talvez deva ser melhor explicada ou deva se tornar mais atrativa para as crianças menores.

Entrevistado 2: Estórias de usuários eu acredito que pode ser modificada a sua aplicação de acordo com a aceitação dela pelas crianças.

3. Alguma coisa mudou nos processos de vocês durante a execução do TCC 2?

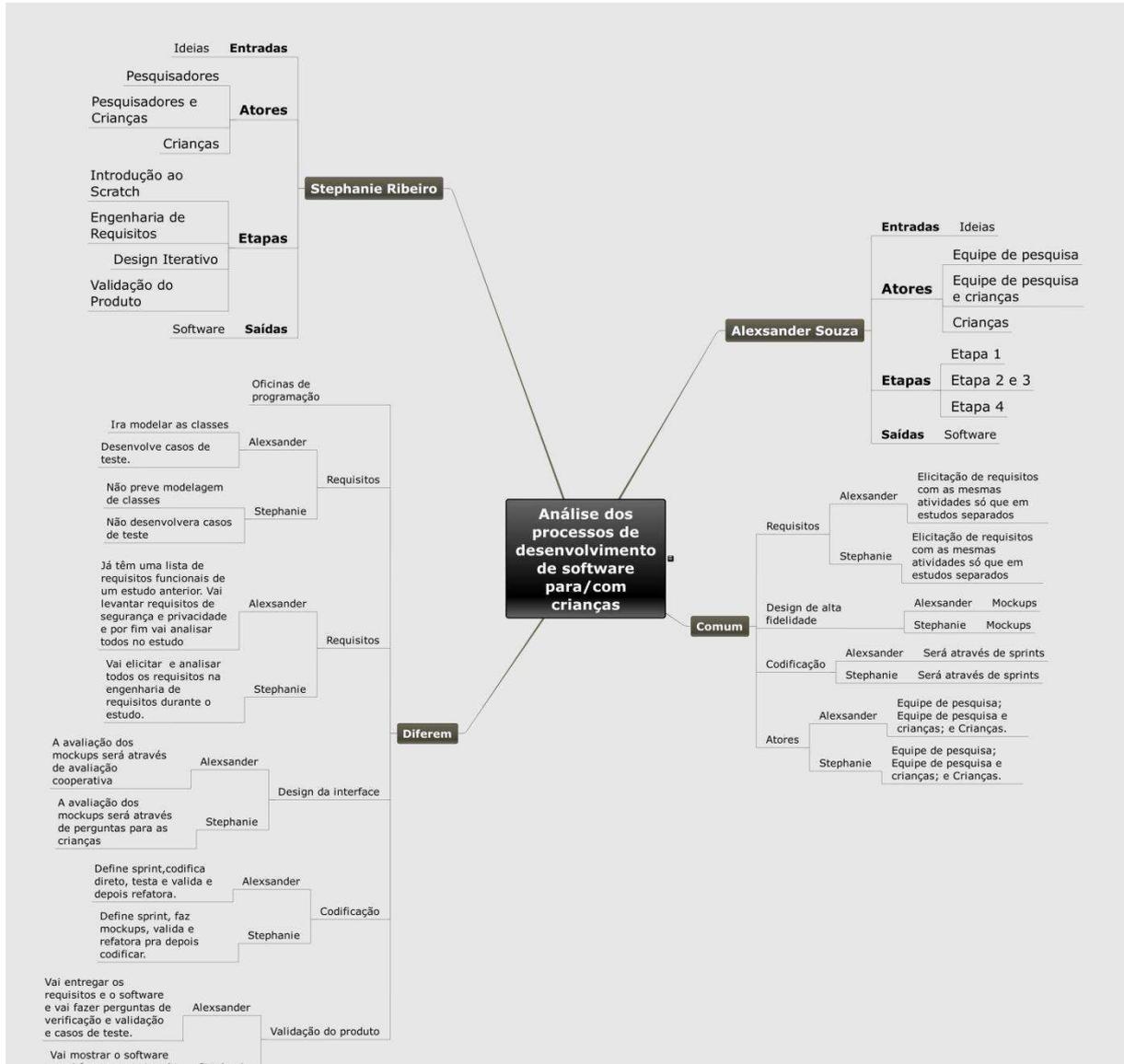
Respostas:

Entrevistado 1: Validação mudou. (Software + Estórias + Testes ) (Esse sistema representa essas estórias?) (Tem tudo aqui?)

Entrevistado 2: Não.

## APÊNDICE G – Mapa mental da análise dos processos de desenvolvimento de software para/com crianças

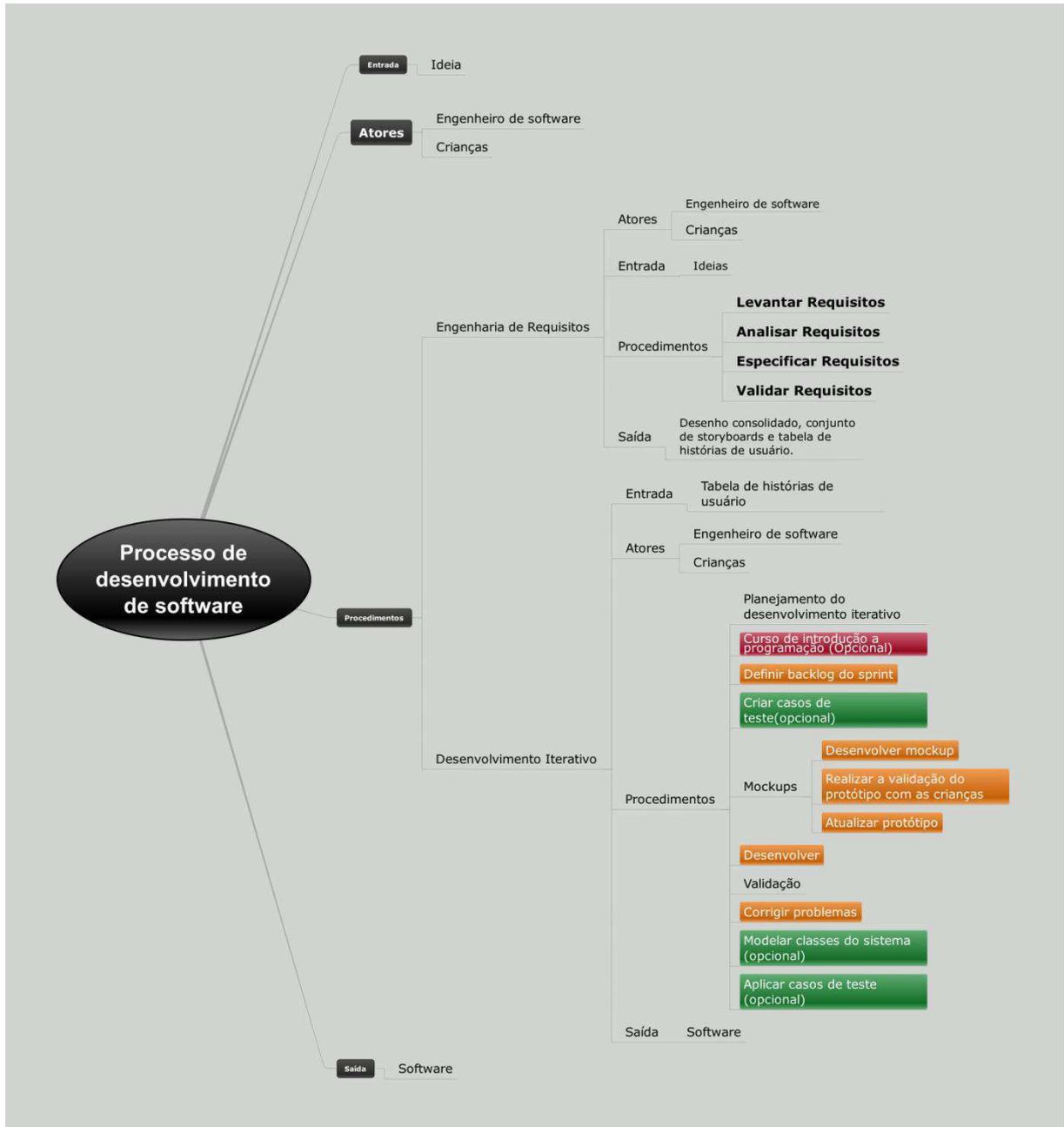
Figura 27 – Mapa mental da análise dos processos de desenvolvimento de software para/com crianças



Fonte: Autoria própria

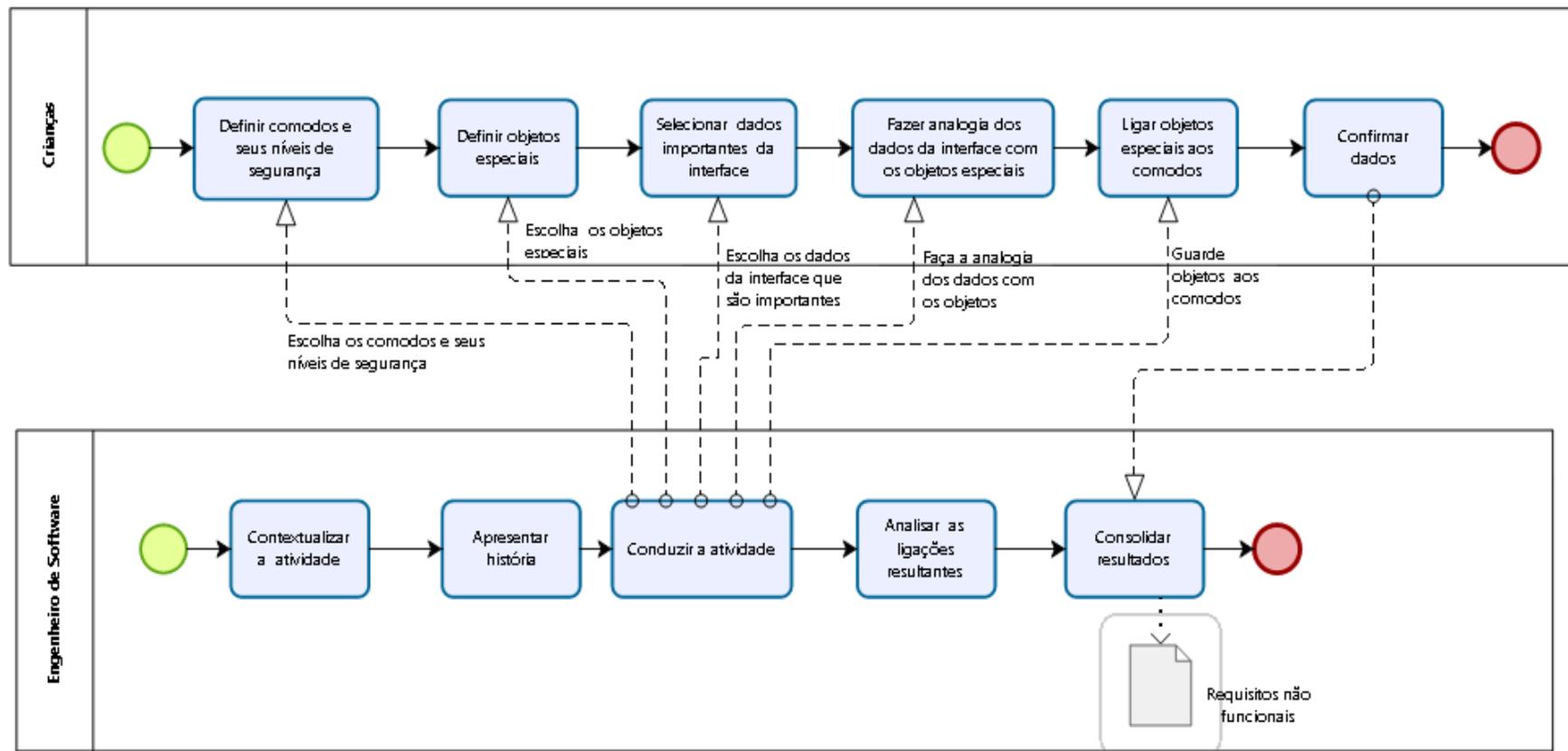
## APÊNDICE H – Mapa mental de proposta do processo de desenvolvimento de software

Figura 28 – Mapa mental de proposta do processo de desenvolvimento de software



## APÊNDICE I – Modelo do subprocesso Analogia Participativa

Figura 29 – Modelo do subprocesso Analogia Participativa



Fonte: Autoria própria

## ANEXOS

### ANEXO A – Agenda do Estudo de Caso 1

Quadro 5– Calendário no PROJOVEM-Adolescente (LÔBO, 2016)

Dia	Atividade(s)
29/10	Explicar a proposta de trabalho Solicitar o consentimento de participação
03/11	O quê? Brainstorming, Braindraw
05/11	O quê? Estórias de Usuário
10/11	Validação de Estórias de Usuário
13/11	O quê? Storyboard
17/11	Avaliação de Mock-ups (wireframe). Produção de Mock-ups (layout).
20/11	Avaliação de Mock-ups (protótipo de alta fidelidade não funcional)
03/12	Avaliação de Mock-ups (protótipo de alta fidelidade funcional com a técnica de Avaliação Cooperativa?)
10/12	Organização para divulgar os resultados do Estudo de Caso.

Fonte: Lôbo (2016)

## ANEXO B – Agenda do Estudo de Caso 2

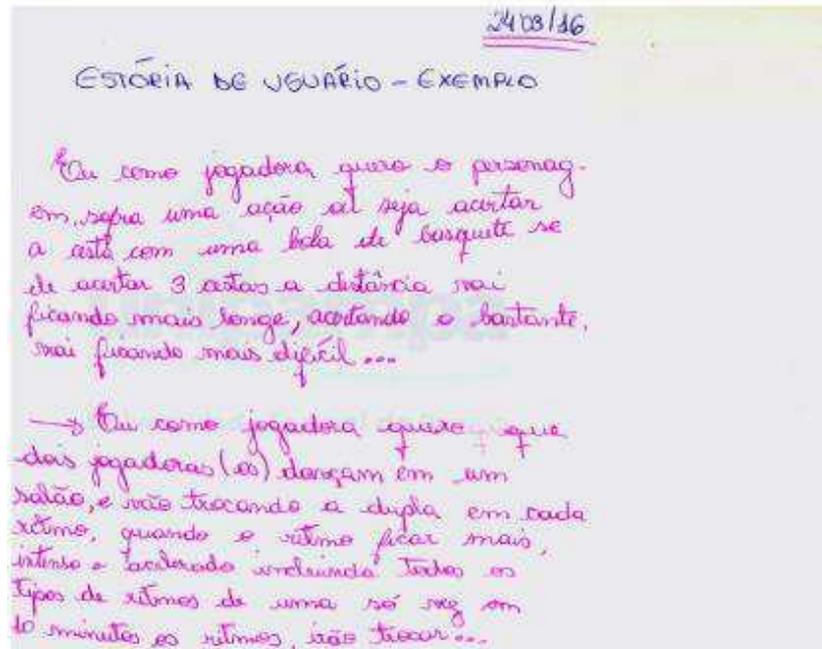
Quadro 6– Calendário na Moradia Transitória (LÔBO, 2016)

Dia	Atividade(s)
08, 10, 15 e 17/03	Explicar a proposta de trabalho Solicitar o consentimento de participação Scratch: propósito de adotá-lo e aprendê-lo, produção de animação, apresentação de possibilidades
22/03	O quê? Brainstorming, Braindraw
24/03	O quê? Estórias de Usuário
29/03	Validação de Estórias de Usuário
31/03	O quê? Storyboard
05/04	Avaliação de Mock-ups (wireframe). Produção de Mock-ups (layout).
07/04	Produção de Mock-ups (protótipo de baixa fidelidade funcional).
12/04	Avaliação de Mock-ups (protótipo de alta fidelidade não funcional) <b>ADIADO PARA 14/04</b>
<b>14</b> , 19, 26/04	Design Iterativo com Scratch (protótipo de alta fidelidade funcional, avaliação iterativa, ...)
28/04	Organização para divulgar os resultados do Estudo de Caso.

Fonte: Lôbo (2016)

## ANEXO C – Exemplo de Lista e Tabela de histórias de usuário

Figura 30 – Lista de histórias de usuário (LÔBO, 2016)



Fonte: Lôbo (2016)

Quadro 7 – Tabela de Histórias de usuário (LÔBO, 2016)

Ator	Operador	Ação	Funcionalidade
Usuário	Devo	Selecionar opção	Escolher atividade/minigame
Usuário	Quero	Selecionar cenários	Mudar cenários
Usuário	Gostaria	Desenhar	Criar cenários
Usuário	Quero	Selecionar características	Criar personagem
Usuário	Quero	Selecionar roupas	Customizar personagem
Usuário	Quero	Selecionar acessórios	Customizar personagem
Usuário	Devo	Selecionar tipo de balão	Indicar o tipo de ideia expressa (pensamento, fala etc.)
Usuário	Gostaria	Selecionar personagem e diferentes posições	Movimentá-lo
Usuário	Quero	Selecionar características do par	Criar par
Usuário	Gostaria	Selecionar música	Mudar música do cenário

Fonte: Lôbo (2016)