



ANÁLISE DE FERRAMENTAS DE SIMULAÇÃO COM FOCO NO PROCESSO DE PRIORIZAÇÃO DE REQUISITOS

Trabalho de Conclusão de Curso

Engenharia da Computação

Bruna Vitória Mota Cruz
Orientadora: Prof. Maria Lencastre



**Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação**

BRUNA VITÓRIA MOTA CRUZ

**ANÁLISE DE FERRAMENTAS DE
SIMULAÇÃO COM FOCO NO
PROCESSO DE PRIORIZAÇÃO DE
REQUISITOS**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Junho de 2017.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 7 de Julho de 2017, às 8:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **BRUNA VITORIA MOTA CRUZ**, orientado pelo professor **Maria Lencastre Pinheiro de Menezes Cruz**, sob título **ANÁLISE DE FERRAMENTAS DE SIMULAÇÃO COM FOCO NO PROCESSO DE PRIORIZAÇÃO DE REQUISITOS**, a banca composta pelos professores:

Sérgio Mario Lins Galdino

Maria Lencastre Pinheiro de Menezes Cruz

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 8,0 (0100)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

Sérgio Mario Lins Galdino

SÉRGIO MARIO LINS GALDINO

Maria Lencastre Pinheiro de Menezes Cruz

MARIA LENCASTRE PINHEIRO DE MENEZES CRUZ

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

*Dedico esse trabalho aos meus pais Elba e Carlos.
E ao meu querido esposo Lyniker.*

Agradecimentos

Agradeço primeiramente ao meu Deus e Salvador Jesus Cristo, por me conduzir a cada dia em sua presença, por ser minha força e coragem para perseverar.

À minha família de forma geral, e particular ao meu esposo Lyniker Nery, pelo apoio, força e principalmente por sempre acreditar em mim. A minha mãe Elba, que sempre esteve ao meu lado compartilhando tanto das alegrias quanto das tristezas, incentivando-me sempre.

Aos companheiros de curso, Fábio Alexandre e Luiza Freire, pelo incentivo e companheirismo, por compartilharem suas experiências, ideias e conselhos.

Aos professores que não pouparam esforços para nos proporcionar a oportunidade de atualizar e expandir nossos conhecimentos.

Resumo

Apesar do notável espaço que o software tem alcançado nas empresas nos diversos ramos, ainda existem muitos desafios relacionados ao seu desenvolvimento. Atualmente, grande parte das empresas não faz a utilização do processo de priorização de requisitos, ou realizam-no de forma tardia, no desenvolvimento de um sistema. Entre as razões associadas à inexistência ou retardo dessa atividade está o fato de algumas organizações acreditarem que a aplicação de técnicas de priorização de requisitos é inútil, cara e requer muita alocação de tempo. Essa atitude, porém, pode ocasionar prejuízos e frustrações, pelo fato do sistema ter uma grande probabilidade de não atender às reais necessidades do cliente. Este trabalho de conclusão de curso evidencia a importância das fases iniciais do desenvolvimento de software, a Engenharia de Requisitos. Ele avalia o uso de algumas ferramentas de simulação computacional para apoiar a tomada de decisão, quando da definição de estratégias para a priorização de requisitos de um software. Entre as ferramentas de simulação, levantadas durante o estudo, três foram escolhidas para a avaliação por se mostrarem disponíveis para *download*: o Arena, o ProModel e o Bizagi Modeler. Para avaliar cada uma das ferramentas, foi aplicada a técnica de priorização de requisito \$100 nas três ferramentas e chegou-se à conclusão sobre qual(is) dela(s) atende(m) de forma mais adequada ao domínio da priorização de requisitos.

Abstract

Despite the remarkable space that software has achieved in companies of various branches, there are still many challenges related to its development. Currently, most companies do not use the requirements prioritization process, or do so late in developing a system. Among the reasons associated with the lack or delay of this activity is the fact that some organizations believe that the application of requirements prioritization techniques is useless, expensive and requires a lot of time allocation. This attitude, however, can cause losses and frustrations, because the system has a high probability of not meeting the real needs of the client. This completion of course work evidences the importance of the early phases of software development, the Requirements Engineering. He evaluates the use of some computational simulation tools to support decision making when defining strategies for prioritizing software requirements. Among the simulation tools, raised during the study, three were chosen for evaluation because they are available for download: Arena, ProModel and Bizagi Modeler. In order to evaluate each one of the tools, the prioritization technique of \$ 100 requirement was applied in the three tools and the conclusion was reached on which one (s) suits the requirements prioritization domain.

Índice de Figuras

Error! Reference source not found. Visão simplificada de um processo de desenvolvimento de software.....	24
Figura 2 Atividade de engenharias de requisitos.....	26
Figura 3 Estrutura Hierárquica Básica.....	34
Figura 4 Formas de estudar um sistema.....	38
Figura 5 Relacionamento dos componentes da simulação	40
Figura 6 Área de trabalho – Arena	42
Figura 7 Barra de comandos geral –Arena	43
Figura 8 <i>Template Basic Process</i> –Arena	43
Figura 9 Área de Trabalho – ProModel.....	45
Figura 10 Barra de comandos geral –ProModel	45
Figura 11 Barra de comandos básicos –ProModel.....	46
Figura 12 Áreas para modelagem –ProModel.....	46
Figura 13 Área de simulação –Bizagi Modeler.....	49
Figura 14 Técnica \$100 – Arena	51
Figura 15 Durante processo de simulação –Arena.....	53
Figura 16 Final simulação –Arena.....	53
Figura 17 Técnica \$100 – ProModel	55
Figura 18 Recursos utilizados – ProModel	55
Figura 19 Técnica \$100 – software Bizagi Modeler.....	56
Figura 20 Durante a simulação – software Bizagi Modeler	56

Índice de Quadros

Quadro 1	Algumas normas, padrões e modelos que buscam qualidade ¹ , focados em produto ou processo de software	22
Quadro 2	Descrição dos passos para aplicação da técnica \$100	29
Quadro 3	Distribuição de unidades aos requisitos	29
Quadro 4	Ranking da priorização de requisitos	29
Quadro 5	Critérios de Qualidade do Cliente + Priorização.....	32
Quadro 6	Critérios de Aceitação + Priorização.....	33
Quadro 7	Componentes relacionados a simulação e seus significados.....	40
Quadro 8	Funções básicas do Promodel	47
Quadro 9	Componentes do simulado BPMN	50

Índice de Tabelas

Tabela 1	Escalas de valor para julgamentos paritários.....	35
Tabela 2	Relatório de simulação do cenário 1 no Arena	54
Tabela 3	Relatório de simulação do cenário 2 no ProModel	56
Tabela 4	Relatório de simulação do cenário 1 no Bizagi Modeler	57

Sumário

Capítulo 1 Introdução	12
1.1 Contextualização da área	12
1.2 Motivação para pesquisa	14
1.3 Caracterização do Problema	15
1.4 Objetivos	15
1.5 Metodologia Usada	16
1.6 Estrutura do Trabalho	17
Capítulo 2 Fundamentação sobre a Engenharia de Software	18
2.1 Engenharia de Software	18
2.2 Qualidade de Software	19
2.3 Processo de Desenvolvimento de Software	21
2.4 Fundamentos da Engenharia de Requisitos	24
2.4.1 Priorização de Requisitos	26
2.5 Técnicas de Priorização	27
2.5.1 Dólar-cem (\$100)	27
2.5.2 MoSCoW	29
2.5.3 Analytic Hierarchy Process (AHP)	32
Capítulo 3 Simulação Computacional	35
3.1 Modelagem e Simulação de Sistemas	35
3.1.1 Componentes da Simulação	39
3.2 Vantagens e Desvantagens do Uso da Simulação	38
3.3 Classificação dos Modelos de Simulação	40
Capítulo 4 Análise das Ferramentas de Simulação Computacional	41
4.1 Software para Simulação	44
4.2 Simulação nas Ferramentas	50
4.2.1 Aplicando a técnica \$100 na ferramenta Arena	51
4.2.2 Aplicando a técnica \$100 na ferramenta ProModel	54
4.2.3 Aplicando a técnica \$100 na ferramenta Bizagi Modeler	55
4.3 Análise das Ferramentas e Considerações	56
Capítulo 5 Considerações Finais	60
5.1 Conclusão	59
5.2 Trabalhos Futuros	60
Bibliografia	62

Capítulo 1

Introdução

Este capítulo faz uma abordagem inicial sobre este trabalho, detalhando a motivação para pesquisa, a caracterização do problema, o objetivo a ser atingido e a estrutura do trabalho.

1.1 Contextualização da área

Na sociedade atual é notável a necessidade das organizações investirem na inovação tecnológica. Não se aplica mais o pensamento de que apenas grandes empresas se preocupam em ter um ambiente totalmente tecnológico. Neste mercado competitivo, onde todos almejam melhores estratégias e mais qualidade, é evidente o impacto e as transformações que a Tecnologia da informação pode provocar no ambiente empresarial, assumindo um papel de vital importância para aumentar a competitividade das organizações (Rodrigues; Pinheiro,2005).

Uma das fases mais importantes no desenvolvimento de sistemas é a fase da Engenharia de Requisitos (ER); os requisitos são a base para todas as atividades do processo de desenvolvimento de software. Logo, definir requisitos de forma coerente e que atenda às reais necessidades dos usuários do sistema é um elemento crítico na garantia de qualidade de software (SOMMERVILLE, 1997).

Para muitos profissionais envolvidos em projetos de software, tanto fornecedores quanto clientes, ao se tratar de escopo, não há limitações e sempre se deseja que o sistema contemple tudo. Mas, não se é possível se desenvolver todas as funcionalidades de um sistema de uma só vez. Uma das formas para se decidir o que dever ser feito com os recursos disponíveis é priorizar os requisitos. Em outras palavras, seria criar uma lista de requisitos, definir uma prioridade para cada um; os mais prioritários ocupam o início da fila (Plínio, 2015). Após priorizados, é definida a ordem para os requisitos com a mesma prioridade.

Porém, muitas vezes, a priorização é feita tardiamente no processo de desenvolvimento de software. Estudos apontam que 53% dos projetos finalizados não correspondem à expectativa dos clientes (José Maurício, 2015). Karlsson e Ryan (1996), afirmam que um dos maiores riscos enfrentados, por organizações que desenvolvem *software*, está relacionado ao não atendimento das necessidades e expectativas dos usuários. Para estes autores, esse risco pode ocasionar danos na reputação da empresa, perda de pedidos e redução dos lucros.

Em um período de contínua mudança no ambiente de negócios de forma global, as organizações estão encontrando cada vez mais dificuldades em lidar e ajustar-se às novas exigências. Segundo Pohl e Rupp (2011), o impacto da Engenharia de Requisitos no desenvolvimento de sistemas de sucesso e focados no cliente não pode mais ser ignorado; ainda segundo os autores, tem se tornado uma prática comum destinar recursos à engenharia de requisitos.

Com as limitações de tempo e recursos, fez-se necessário garantir que os requisitos de maior importância sejam realizados o mais rápido possível. De tal forma que os requisitos já documentados devem ser priorizados. Segundo Pohl e Rupp (2011), priorizar requisitos é ordenar requisitos de maior importância em relação a um ou vários critérios de priorização.

Porém, a priorização de requisitos não deve ser feita de qualquer forma, devem-se considerar várias questões relacionadas ao projeto sendo desenvolvido. Existem várias técnicas e ferramentas que dão suporte à priorização de requisitos; em Achimugu (2014), são mapeadas 49 técnicas. Algumas se adaptam mais a sistemas complexos enquanto outras são mais simples e fáceis de serem executadas. Pohl e Rupp (2011), relata que cada técnica de priorização é (pelo menos parcialmente) baseada nos julgamentos e avaliações subjetivas das partes interessadas, isto é, dos *stakeholders*.

A realização de priorização de requisito, assim como a escolha da melhor técnica, pode ser algo complexo e levar tempo. Dessa forma é importante avaliar o custo-benefício de seu uso, e principalmente da estratégia de sua aplicação.

A abordagem via simulação implica na utilização de um programa computacional que executa uma série de instruções que após processadas transmitem ao usuário a sensação que o modelo representa, de fato, o sistema em que está fundamentado (Freitas Filho, 2008).

Uma das vantagens do estudo e análise através da simulação é que, uma vez desenvolvido um modelo de simulação válido, pode-se incorporar modificações (explorar novas políticas, procedimentos, métodos, etc.) sem se preocupar com os custos efetivos ou a perturbação do sistema real (Banks, 1999). A simulação permite acelerar ou desacelerar os fenômenos para que estes possam ser investigados. Uma vez desenvolvido um modelo de simulação válido, pode-se explorar políticas, procedimento operacionais ou métodos, com um baixo custo, sem a necessidade de se interferir/perturbar o sistema real. Assim, torna-se viável diagnosticar problemas e compreender a importância de determinadas variáveis, como também, aumentar a compreensão do desempenho do sistema no geral (Banks, 1999).

1.2 Motivação para pesquisa

A motivação deste estudo se encontra nas vantagens que as ferramentas de simulação podem trazer na tomada de decisões nas fases iniciais no desenvolvimento de sistemas, e de ainda existir muito pouco realizado na área de priorização de requisitos nesse contexto (Timóteo, 2016). Acredita-se que a simulação pode facilitar o planejamento e aplicação da priorização de requisitos, uma vez que permite avaliar várias questões como melhor técnica a usar, tempo que será despendido, etc. Dessa forma, poderá se promover e justificar o uso das técnicas de priorização, aumentando conseqüentemente a qualidade dos produtos finais, uma vez que estes têm grande probabilidade de atender melhor às expectativas dos clientes.

Existem diferentes ferramentas de suporte à simulação que auxiliam na avaliação e execução de diferentes cenários de um modelo de simulação. Porém, não há nenhuma ferramenta que auxilie a simulação na Engenharia de Requisitos e que atenda à análise e decisões no processo de priorização de requisitos. Na

dissertação de Timóteo (2016), ele propõe a conceitualização de uma ferramenta de Simulação de Domínio Específico para Priorização de Requisitos que também serviu de motivação para este trabalho. Foi possível perceber que seria interessante identificar as ferramentas de simulação que ajudassem na construção de modelos para aplicar técnicas da priorização de requisitos.

1.3 Caracterização do Problema

Este trabalho de conclusão de curso foca na complexidade em realizar a escolha da técnica de priorização de requisitos adequada, considerando-se fatores como, por exemplo: tempo da priorização, *stakeholders*, custo, entre outros.

1.4 Objetivos

Este projeto tem por objetivo, considerando o domínio da priorização de requisitos de software, analisar ferramentas de simulação existentes para identificar qual (is) permite (m) uma boa forma de representação e análise do domínio considerado assim como questões de gratuidade da ferramenta. Aumentando a probabilidade de que o sistema desenvolvido, com ajuda dessas ferramentas de simulação, satisfaça as reais necessidades dos usuários envolvidos.

Os sub-objetivos deste trabalho serão:

- a) Levantar conceitos relacionados ao processo de priorização de requisitos e simulação;
- b) Levantar conjunto de ferramentas de simulação a serem estudadas;
- c) Definir critérios para avaliação das ferramentas;
- d) Definir modelos e experimentos a serem usados;
- e) Projetar modelos e rodar experimentos nas ferramentas de simulação;
- f) Definir qual (is) a (s) ferramenta (s) de simulação atende (m) aos critérios definidos para o domínio da priorização de requisitos.

1.5 Metodologia Usada

Para alcançar os objetivos propostos foram seguidas as seguintes etapas

1. Estudo da simulação;
2. Levantamento das ferramentas de simulação disponíveis;
3. Estudo de priorização de requisitos e de técnicas de priorização;
4. *Download* de versões de ferramentas de simulação que são gratuitas;
5. Seleção de técnicas de priorização, onde foi escolhida a \$100;
6. Entrevista informal com um engenheiro de requisitos de uma empresa de médio porte do Porto Digital, para auxílio da montagem do modelo de simulação referente a um processo de priorização;
7. Aplicação da técnica de priorização \$100 nas ferramentas de simulação escolhidas (Arena, ProModel, Bizagi Modeler);
8. Análise de resultados.

Existem as ferramentas para construção de modelos para simulação que podem ser usados em aplicação geral e os que são usados em áreas específicas. Para este trabalho de conclusão de curso as ferramentas de simulação escolhidas são de propósito geral, com o objetivo de avaliar qual melhor se encaixa no domínio da priorização de requisitos.

Outras estratégias definidas para escolher as ferramentas de simulação a serem avaliadas, foi considerar a existência de uma versão gratuita da ferramenta e a sua disponibilidade para *download*, possuírem um mínimo de documentação para entendimento da ferramenta e ser utilizada por um grande número de empresas.

Com relação à estratégia para escolha da técnica de priorização a ser usada, foi avaliado o fato de ser simples e conhecida.

1.6 Estrutura do Trabalho

Além deste capítulo introdutório, este trabalho está dividido em mais 4 capítulos descritos a seguir:

- Capítulo 2 – Fundamentação sobre a Engenharia de Software: apresenta conceitos relacionados à engenharia de software, qualidade, processo de Engenharia de Requisitos, e priorização.
- Capítulo 3 – Simulação Computacional: apresenta os conceitos relacionados à simulação computacional, vantagens e desvantagens, formas de classificação dos modelos de simulação.
- Capítulo 4 - Análise das Ferramentas de Simulação Computacional e Aplicação da técnica \$100: apresenta para cada ferramenta escolhida como foi aplicada a técnica \$100.
- Capítulo 5 – Considerações Finais: apresenta a conclusão do trabalho e propõe trabalhos futuros.

Capítulo 2

Fundamentação sobre a Engenharia de Software

Este capítulo faz uma breve descrição sobre o conceito da Engenharia de Software, qualidade de software, e processo de desenvolvimento de software. Em seguida, detalha o processo da Engenharia de Requisitos, onde ocorre a priorização. Por fim, apresenta técnicas de priorização de requisitos.

2.1 Engenharia de Software

Em uma conferência realizada pela *Association for Computing Machinery* (ACM) nos anos 70, Dijkstra (1972) cunhou o termo “crise de software” e desde então o processo de desenvolvimento de software evoluiu significativamente. Nesta conferência, Dijkstra manifestou as dificuldades em relação aos orçamentos, prazos de entregas estourados, requisitos não atendidos, baixa qualidade e manutenibilidade do software. As possíveis soluções para resolver esses problemas estariam na melhoria das metodologias, técnicas e ferramentas, aumento na educação sobre o “que é” e “como fazer” o desenvolvimento de software.

O termo Engenharia de Software foi criado na década de 1960 e utilizado oficialmente em 1968 na *NATO Science Committee* (*comitê de ciência da OTAN*). Sua criação surgiu numa tentativa de contornar a crise de software e dar um tratamento de engenharia (mais sistemático e controlado) ao desenvolvimento de software complexos.

Segundo Pressman, (2011) para desenvolver software que esteja preparado para enfrentar os desafios do século vinte e um, deve-se conhecer alguns fatos reais:

- O software tornou-se profundamente incorporado em praticamente todos os aspectos da vida das pessoas, e as pessoas interessadas nos recursos e funções oferecidas por uma aplicação tem crescido no mundo todo.
- Os requisitos tecnológicos demandados por indivíduos, empresas e órgão governamentais estão mais complexos a cada ano.
- As pessoas, negócios e governos estão, de forma crescente, dependentes de software para tomarem as suas decisões e controlar seu cotidiano. Uma falha nesse software pode levar desde pequenas falhas até a falhas catastróficas.
- À medida que uma aplicação aumenta sua base de usuários e longevidade também crescem; assim, o software deve ser passível de manutenção.

Uma primeira definição de Engenharia de Software, proposta por Fritz Bauer, é o estabelecimento e o uso de princípios sólidos de engenharia para que se possa obter economicamente um software que seja confiável e funcione eficientemente em máquina reais (Pressman, 2006).

A norma IEEE93a propõe uma outra definição, afirmando que a Engenharia de Software é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação da engenharia ao software.

A Engenharia de Software no geral envolve processos, métodos e ferramentas que permitem construir sistemas complexos baseados em computador, dentro do prazo determinado e com qualidade.

2.2 Qualidade de Software

O alto grau de importância que o software assume no dia a dia das organizações fez com que a busca por produtos de software de qualidade crescesse. Assim, fez-se necessário que alguns critérios relacionados a qualidade de software fossem definidos.

O que seria qualidade de software? É comum encontrar várias definições onde o conceito do software está relacionado a atender ou não aos requisitos, ou seja, às necessidades e expectativas dos usuários. Algumas definições simples são:

- Qualidade é estar em conformidade com os requisitos dos clientes;
- Qualidade é antecipar e satisfazer os desejos dos clientes;
- Qualidade é escrever tudo o que se deve fazer e fazer tudo que foi escrito;

Em seu livro Pressman (2006), define qualidade de software como conformidade com requisitos funcionais e de desempenho explicitamente declarados, padrões de desenvolvimento claramente documentados e características implícitas que são esperadas de todos software profissionalmente desenvolvido. Apesar dos modelos aplicados na garantia da qualidade de software atuarem, em especial, no processo, o principal objetivo é garantir um produto final que satisfaça às expectativas do cliente, dentro daquilo que foi acordado inicialmente.

A qualidade do produto está relacionada com o seu desempenho, falta de falhas ou erros. Ela pode ser medida pela relação entre a eficiência e a eficácia, um produto que atenda aos requisitos necessários para o cliente (Weber, 2005).

Porém, um aspecto interessante da qualidade de software é que além dela existir deve ser reconhecida pelo cliente. Por isso, fez-se necessário existir algum tipo de certificação oficial e que fosse emitida com base em um padrão. O Quadro 1 demonstra algumas normas e padrões que foram desenvolvidos na busca pela qualidade, com base em produto ou processo de software.

Quadro 1: Algumas normas, padrões e modelos que buscam qualidade, focados em produto ou processo de software

Nome	Descrição
Norma ISO/IEC 9126 (NBR 13596)	Define as características de qualidade de software que devem estar presentes em todos os produtos (funcionalidade, confiabilidade, desempenho, usabilidade, manutenibilidade e portabilidade).
Norma ISO/IEC 12119	Estabelece os requisitos de qualidade para pacotes de software;
Norma ISO/IEC 14598-5	Define um processo de avaliação da qualidade de produto de software.
Norma ISO/IEC 12207	Define um processo de ciclo de vida de software.
Norma ISO/IEC 9000-3	Apresenta diretrizes para a aplicação da ISO 9001 por organizações que desenvolvem software ao desenvolvimento, fornecimento e manutenção.
Norma ISO/IEC 15504	Aprovada como norma em 2003 é focada na avaliação de processos organizacionais, voltada para a qualidade de processo e níveis de capacidade.
Norma ISO 9001	Define um modelo para garantia de qualidade em Projeto, desenvolvimento, instalação e assistência técnica (processo).
CMM	<i>Capability Maturity Model</i> - descreve os principais elementos de um processo de software efetivo, descreve também os estágios através dos quais organizações de software evoluem quando definem, implementam, medem, controlam e melhoram seus processos de software.
CMMI	Capability Maturity Model Integration – Integração de Modelos de Maturidade e de Capacidade. É uma evolução do CMM, contém algumas práticas para o processo de desenvolvimento de software.
MPS.Br	Melhoria de Processo do Software Brasileiro - Define aprimoramentos para um modelo de melhoria e avaliação de processo de software e serviços.

Fonte: (A autora)

Para alcançar a qualidade, além da aplicação das normas e padrões, a conformidade com os requisitos é uma questão bastante relevante para que se alcance a qualidade de software (Cordeiro, 2010). Isto pode ser percebido através das definições de qualidade de software citadas anteriormente.

2.3 Processo de Desenvolvimento de Software

As empresas estão se tornando mais dependentes dos sistemas de informações. A cada dia, diversos modelos, ferramentas e propostas têm sido projetadas, desenvolvidas e sugeridas, visando permitir que as empresas se capacitem evolutivamente para o projeto de software.

O CMM é um modelo utilizado para mediar a maturidade de uma organização nos processos de desenvolvimento de software. Ele é um modelo gerencial, que organiza as melhores práticas existentes. Quanto maior o controle sobre o processo de desenvolvimento de software, mais madura se tornará a organização.

Segundo Moreira (2011), o processo de desenvolvimento de software ou processo de software, corresponde ao conjunto de atividades relacionadas que são desempenhadas pelos desenvolvedores desde concepção até a liberação do produto.

Alguns objetivos do processo de desenvolvimento incluem a definição:

- Das atividades a serem executadas;
- De quando determinada atividade dever ser executada;
- A pessoa ou grupo de pessoas que deve executar tais atividades;
- Da padronização no processo de desenvolvimento;

Elaborar um processo de desenvolvimento de software significa determinar, de forma precisa e detalhada, quem faz o que, quando e como. Um processo pode ser visto como uma instância de um método com suas técnicas e ferramentas associadas, elaborado durante a etapa de planejamento, no qual as atividades foram alocadas aos membros da equipe de desenvolvimento, com prazos e métricas definidas para avaliar com elas estão sendo realizadas.

De uma forma geral, um projeto de desenvolvimento de software tem início quando se identifica a necessidade de desenvolvimento e decide-se investir no mesmo (Booch et al.,2000). A partir daí são realizadas diversas atividades em diferentes etapas, dentre as quais é possível destacar:

- Análise de Requisitos;
- Análise e Projeto;
- Implementação;
- Testes;

Uma das metodologias que atualmente mais tem sido usada na construção de software é o desenvolvimento iterativo. Nesta abordagem o desenvolvimento é composto por uma série de miniprojetos, que serão chamados de iterações. Em cada iteração, são realizadas as atividades que já foram listadas. O objetivo de cada interação é obter um produto que seja de qualidade superior ao produto derivado da iteração anterior. Ao final de várias iterações será obtido o produto final pronto para uso (Larman, 2004).

De uma forma simplificada, a figura 1 elaborada por Cordeiro e Freitas (2008), ilustra o processo de desenvolvimento de software.

Figura 1 – Visão simplificada de um processo de desenvolvimento de software



. Fonte: Cordeiro e Freitas (2008).

Dentre as etapas citadas, do processo de desenvolvimento de software, a que procura entender a natureza do programa que irá ser desenvolvido é a fase de análise de requisitos. O engenheiro de software ou analista deve compreender o domínio da informação para o software, bem como a função, desempenho e interface exigidos (Pressman, 2010).

Para Karlsson e Ryan (1996), um dos maiores riscos enfrentados por organizações que desenvolvem software comercial está associado ao não atendimento das necessidades e expectativas dos usuários. Esses riscos,

segundo estes usuários, podem ocasionar danos, perda de pedidos e redução de lucro da empresa.

A diferença entre metodologias, que buscam a qualidade do produto de software e as que visam qualidade de processo de desenvolvimento, é que as primeiras são realizadas ao final do processo de desenvolvimento quando o software já se encontra concluído. Já as segundas definem as atividades que devem ser realizadas durante o processo de desenvolvimento de software.

Pressman (2010), afirma que entender os requisitos de um sistema está entre as tarefas mais difíceis enfrentadas por engenheiros de software. Segundo o autor, esta fase é de crucial importância para o sucesso do projeto; por apresentar uma série de dificuldades surgiu um campo dentro da engenharia de software, chamado de Engenharia de Requisitos.

2.4 Fundamentos da Engenharia de Requisitos

Uma das principais causas de falhas nos projetos são os erros na hora de especificar os requisitos. Segundo Leffingwell (2011), garantir a participação do usuário no processo de desenvolvimento, cuidar com requisitos e especificações incompletas, mudanças e atentar para as reais necessidades dos usuários são os grandes desafios da Engenharia de Requisitos.

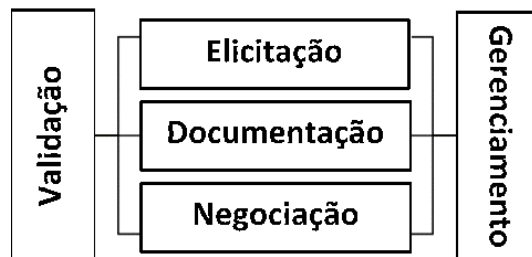
O objetivo principal da Engenharia de Requisitos é conseguir alcançar uma especificação completa do sistema de software. Ela tem por características conter as atividades que estão relacionadas à definição de requisitos (Kotonya; Sommerville, 1998). Para melhorar a comunicação entre clientes e desenvolvedores são feitas anotações e refinamentos dos requisitos, aumentando a transparência do software. Para que um projeto seja dito bem-sucedido é necessário que haja consenso entre clientes e desenvolvedores, que são as partes envolvidas.

A norma IEEE 610 define requisito como uma condição ou capacidade demandada por um usuário para resolver um problema ou atingir um objetivo.

Entende-se também como uma condição ou capacidade que necessite estar presente em um sistema ou em seu componente para satisfazer um contrato ou documento. O requisito tende a ser mais objetivo que a descrição da necessidade e pode ser escrito com a ideia de implementação em mente.

Mesmo que cada empresa ou organização possua um processo específico adaptado às suas características, é possível estabelecer um conjunto de atividades básicas que devem ser consideradas na definição de um processo de Engenharia de Requisitos. Tomando como base as atividades propostas por Klaus Pohl (2010), é considerado que no processo de Engenharia de Requisitos deve contemplar as atividades de: elicitação, documentação, negociação, validação e gerência de requisitos. Ver Figura 2.

Figura 2 Atividade de Engenharia de Requisitos



Fonte: Baseado em (Klaus Pohl, 2010)

- Elicitação: Nesta atividade o engenheiro de software busca entender quais são as necessidades do usuário que devem ser atendidas (Thayer, 1993). Contém a identificação da fonte de requisitos e o levantamento dos novos requisitos.
- Documentação: Os resultados da elicitação são transformados em um documento que organiza os sistemas utilizando linguagem natural e/ou modelos conceituais.
- Validação: Os requisitos que já foram documentados devem ser validados desde o princípio, garantindo os critérios de qualidade que foram previamente definidos.
- Gerenciamento: O gerenciamento de requisitos abrange toda e qualquer medida necessária para estruturar requisitos. Manter a consistência dos

requisitos mesmo após eventuais mudanças (evoluções tecnológicas ou alterações na organização na qual é usado).

2.4.1 Priorização de Requisitos

Os requisitos são demandas dos usuários, que possuem valor para eles. Porém, os recursos necessários para o desenvolvimento de software são limitados devido a restrições de tempo e orçamento, tornando-se difícil implementar todos os requisitos identificados para o sistema. Os requisitos geralmente são implementados em etapas e a priorização ajuda a definir quais devem ser implementados prioritariamente (Allen et al., 2008).

Selecionar o subconjunto de requisitos que serão implementados em uma determinada fase de desenvolvimento do produto é uma atividade complexa e importante para o sucesso do projeto de desenvolvimento (Carlshamre, 2001). Segundo Berander (2004), priorizar requisitos não é restringido apenas a questionar os participantes do projeto sobre quais são as suas prioridades relacionadas aos requisitos do sistema. Existe todo um processo social, no qual considera-se aspectos como o ambiente organizacional, os valores de mercado, como cada membro do projeto se comporta e suas respectivas disponibilidades.

Uma correta priorização de requisitos pode determinar não só o sucesso de um projeto, como também a sobrevivência de pequenas organizações de desenvolvimento de software (Azar; Smitch, 2007).

Segundo Klaus Pohl (2010), são estabelecidas quatro atividades preparatórias para a priorização de requisitos, são elas: selecionar *stakeholders*, selecionar requisitos, selecionar a técnica de priorização a utilizar e definir o critério de priorização. Os critérios que norteiam a priorização de requisitos são as questões corporativas e às técnicas de priorização.

Existe um grande número de técnicas para priorização de requisitos, em Achimugu (2014), são mapeadas 49 técnicas (conforme visto na seção 1.1) saber qual o método mais adequado a ser usado torna-se difícil. É importante ressaltar que nenhuma das técnicas pode ser considerada a melhor, mas é

preciso escolher qual a técnica é mais adequada para cada situação, em termos de estabilidade, precisão e tempo para sua aplicação (Vestola, 2011).

2.5 Técnicas de Priorização

O objetivo da priorização de requisitos é atribuir valores aos requisitos e estabelecer uma ordem relativa entre esses requisitos. As técnicas de priorização podem ser classificadas em grupos de escala: nominal, ordinal e razão (Qiao, 2009). As técnicas que usam:

- Escala nominal, separam os requisitos em grupo de prioridades, onde todos os requisitos de um mesmo grupo têm prioridade igual; MoSCoW é um exemplo desse tipo de técnica ;
- Escala ordinal têm por objetivo ordenar uma lista de requisitos; *Bubble Store* é um exemplo desse tipo de técnica,
- Escala de proporção, produz resultados que trazem diferenças relevantes entre os requisitos; exemplos desse tipo de técnica são o *Hundred Dollar* (\$100) e a *Analytic Hierarchy Process* (AHP).

A seguir, serão detalhadas 3 técnicas de priorização de requisitos: Dólar-cem (\$100), MoSCoW e AHP. Essas técnicas foram escolhidas por serem as mais utilizadas nos trabalhos pesquisados durante o desenvolvimento desse trabalho de conclusão de curso.

2.5.1 Dólar-cem (\$100)

É uma técnica simples, em que são fornecidos 100 dólares às partes interessadas, isto é, aos *stakeholders* (Berander, 2004). Cada *stakeholder* fica responsável por distribuir os 100 dólares entre os requisitos a serem priorizados. Em seguida, é feita a priorização. O resultado obtido indica quanto um requisito é mais importante de que os outros em termos proporcionais. O Quadro 2 apresenta os passos da aplicação da técnica dólar-cem, baseada na descrição dada por Leffingwell e Widrig (2003).

Quadro 2: Descrição dos passos para aplicação da técnica \$100

Descrição
1. Tomar conhecimento dos requisitos, a fim de entendê-los;
2. Distribuição dos 100 dólares disponíveis entre os referidos requisitos;
3. Realização da soma dos dólares, para poder constatar que tal somatório teve um valor menor ou igual a cem.
4. Caso o somatório tenha sido superior a cem, é então realizada a redistribuição dos dólares;
5. Após a distribuição ser concluída, é feita a montagem do <i>ranking</i> , conforme os dólares recebidos por cada requisito.

Fonte: (A autora)

Como exemplo, suponha que uma empresa está desenvolvendo um projeto e que os requisitos levantados foram R1, R2, R3 e R4. A atribuição do valor aos requisitos é apresentada no Quadro 3 e o *ranking* da priorização no Quadro 4.

Quadro 3: Distribuição de unidades aos requisitos

Requisitos	Dólares
R1	10
R2	55
R3	5
R4	30
	Total = 10+55+5+30 = 100

Fonte: (A autora)

Quadro 4: *Ranking* da priorização de requisitos

Requisitos	Dólares	Ranking Final
R2	55	1
R4	30	2
R1	10	3
R3	5	34

Fonte: (A autora)

Um problema com esta técnica surge quando há muitos requisitos para serem priorizados (Berander 2004). Se, por exemplo, existirem 25 requisitos,

há em média 4 dólares para distribuir para cada requisito, muitas vezes a soma dá uma quantidade maior ou menor que 100 dólares.

Já Leffingwell e Widrig (2003) aponta como deficiência desta técnica o fato de que as partes interessadas podem atribuir todos os seus dólares a um único requisito. Isso resulta em obrigação ou em restrição aos interessados, não ocorrendo a priorização de acordo com as necessidades.

2.5.2 MoSCoW

De acordo com Miranda (2011), MoSCoW é uma técnica de priorização onde tradicionalmente se pergunta ao cliente como ele classifica as suas preferências, considerando uma dada categorização. Uma classificação pode ser dividir em:

- Deve Ter (*Must Have*): quando é imprescindível para o escopo do projeto; aquelas funcionalidades da aplicação, que sem elas a aplicação perderia totalmente o sentido.
- Deveria Ter (*Should Have*): quando é importante ser incluído no escopo do projeto, mas não é imprescindível. Funcionalidades que se não desenvolvidas, não farão com que o produto perca o seu valor de negócio.
- Poderia Ter (*Could Have*): quando é bom ter, mas não é importante. É a cereja do bolo, na maioria das vezes, os clientes aceitam comer bolo sem cereja, até porque fica mais barato.
- Não Terá Agora (*Won't Have for Now*): Não geram valor de negócios no momento. Tudo que não será desenvolvido por enquanto, mas podem ser considerados no futuro.

Para ilustrar o uso da técnica, a melhor maneira, é por meio de um exemplo, conforme apresentado a seguir. Em seu treinamento, PRINCE2, Turley (2012) descreve que tem por objetivo criar um MP3 de alta tecnologia que seja fino e de fácil uso para jovens, incluindo todas as características normais de um MP3, com fone de ouvido *Bluetooth* e cartão de memória. A composição do MP3 será basicamente:

- *Bluetooth* MP3 com base em Android;
- Armazenamento de 8 a 16 GB com SD para expansão de memória;
- Tela sensível ao toque [*touchscreen*] de 2.5" e a prova de arranhões;
- Bateria de íons de lítio;
- Conexão USB padrão para transferência de dados e carga de bateria;
- Processador ARM11 620 MHz;
- Suporte a idiomas: 23 línguas.

Para o fone de ouvido *Bluetooth*:

- Fone de ouvido *Bluetooth* estéreo com microfone;
- Versão do *Bluetooth* 2.1.

O Quadro 5 apresenta a priorização dos critérios de Qualidade do Cliente utilizando a técnica MoSCoW, sendo M Precisa Ter (*Must Have*) e S Deve Ter (*Should Have*).

Já o Quadro 6, utilizando da mesma classificação de MoSCoW, apresenta os critérios de aceitação. Segundo OGC (2011), critérios de aceitação devem ser priorizados, pois ajudam caso haja a necessidade de troca de alguns critérios. Os critérios de aceitação formam uma lista priorizada dos atributos necessários para que as parte interessadas aceitem o produto. Quando o produto demonstra que todos os critérios de aceitação foram satisfeitos, o projeto pode ser encerrado. A técnica de priorização MoSCoW é de utilização simples, robusta, flexível e adaptável; ela possibilita atribuir importância relativa aos requisitos.

As informações contidas no Quadro 5 e no Quadro 6 foram coletadas em discussões com as partes interessadas, refinadas e por fim incluídas na descrição do produto. Procurou-se redigir o texto de forma a evitar interpretações erradas e suposições imprecisas sobre os requisitos de

qualidade do projeto, bem como se considera apenas os critérios: *Must Have* (Precisa Ter) e *Should Have* (Deve Ter) (Oliveira, 2014).

Quadro 5: Critérios de Qualidade do Cliente + Priorização

Expectativas de Qualidade do Cliente	MoSCoW
Uso médio – todas as partes – 5 anos (incluindo uso acelerado e envelhecimento do equipamento) – Empresa Externa	M
Fácil de usar para usuários inexperientes (pesquisa com 1000 pessoas)	M
Suporte com base em aplicativos Android (sem suporte GSM ou GPS)	M
Suporte à integração com computadores PC, Linux e Apple	M
Slot SD expansível para suportar cartões de 8, 16 e 32 GB	M
Tela sensível ao toque 2.5" (<i>touchscreen</i>)	M
Vida útil da bateria – utilização normal – 36 horas	M
Carga rápida da bateria – 80% em 30 minutos	S
Conexão USB para conexão com PC e para carga	M
Tela à prova de arranhões – Classe	M
Alcance do <i>Bluetooth</i> – 13 metros	M
<i>Bluetooth</i> Estéreo v2.1	M
Fones: Bateria – 12 horas de utilização	S
Fones de ouvido: com microfone (uso com telefone)	M
Fone de ouvido: alcance do <i>Bluetooth</i> – 13 metros	M

Fonte: Turley (2012).

Quadro 6: Critérios de Aceitação + Priorização

Critérios de Aceitação	MoSCoW
Custos de fabricação da unidade: menor que \$50	M
Custo de Projeto para desenvolver e testar: menor que \$500.000	M
5 Anos – Teste Médio – Com certificado da empresa	M
Fácil de usar para 90% dos usuários	M
Suporte a aplicativos Android (sem suporte GSM ou GPS)	M
Compatível com sistemas Windows, Apple e Linux	M
Slot cartão SD expansível: suporte a cartões de 8, 16 e 32GB	M
Tela sensível ao toque 2.5" (<i>touchscreen</i>)	M
Carga rápida bateria 80% em 30 minutos	S
Suporte a Línguas – 23 línguas	S
Dimensões - 60mm x 60mm x 4 mm (H*W*D)	S
Peso - 28.3 gramas	S

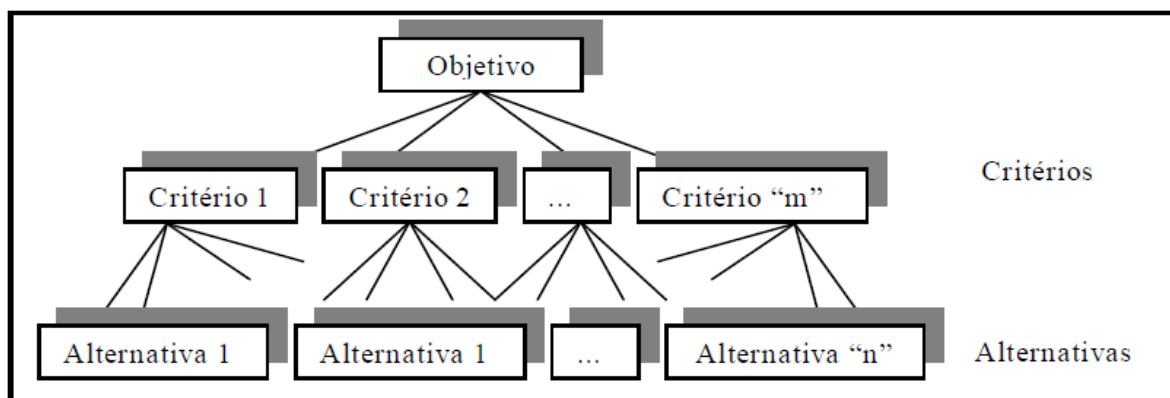
Fonte: segundo Turley (2012).

2.5.3 Analytic Hierarchy Process (AHP)

Proposto por Saaty no início dos anos 70, o método AHP objetiva a seleção/escolha de alternativas em um processo decisório que considere múltiplos critérios. É um método de tomada de decisão que foi adaptado para a priorização de requisitos. AHP é uma técnica estatística sistemática com base na avaliação relativa, que tem sido muito usada para priorizar os requisitos pela comunidade de software (Regnell et al., 2001). Segundo Saaty (2000), o método AHP está estruturado segundo três princípios:

- 1- Construção de hierarquias: Sistemas complexos podem ser melhor compreendidos através do particionamento destes em elementos constituintes, estruturando-os hierarquicamente e, então, sintetizando os julgamentos da importância relativa destes elementos em cada nível da hierarquia em um conjunto de prioridades (Saaty, 2000). A Figura 3 define o foco principal, que seria o objetivo do problema, os critérios e subcritérios e as alternativas.

Figura 3 – Estrutura Hierárquica Básica



Fonte: Adaptado de Saaty (2000)

2- Definição de Prioridades: Segundo Saaty (2000), “o ser humano tem a habilidade de perceber as relações entre as coisas que observa, comparar pares de objetos similares à luz de certos critérios, e discriminar entre os membros de um par através do julgamento da intensidade de sua preferência de um elemento sobre o outro”. É necessário cumprir as seguintes etapas:

- Julgamento paritário: Os elementos serão julgados par a par em um nível superior, compondo as matrizes de julgamento A, através de uma escala de pontuação (1-9), conforme apresentado na Tabela 1.

Tabela 1 - Escalas de valor para julgamentos paritários

Escala Verbal	Escala Numérica
Igual preferência (importância)	1
Preferência (importância) fraca	3
Preferência (importância) moderada	5
Preferência (importância) forte	7
Preferência (importância) absoluta	9
Julgamentos intermediários	2,4,6,8

Fonte: Adaptado de Saaty (2000)

A quantidade de julgamentos necessários para a construção de uma matriz de julgamentos genérica A é $n(n-1)/2$, onde n é o número de elementos pertencentes a esta matriz (Saaty, 2000). Os elementos de A são definidos pelas condições:

$$A = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{21} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ 1/a_{n1} & 1/a_{n2} & \cdots & 1 \end{bmatrix}, \text{ onde:}$$

$$a_{ij} > 0 \Rightarrow \textit{positiva}$$

$$a_{ij} = 1 \therefore a_{ji} = 1$$

$$a_{ij} = 1/a_{ji} \Rightarrow \textit{recíproca}$$

$$a_{ik} = a_{ij} \cdot a_{jk} \Rightarrow \textit{consistência}$$

- Normalização das matrizes de julgamento: Após a soma dos elementos serão obtidos quadros normalizados de cada coluna das matrizes de julgamento e posterior divisão de cada elemento destas matrizes pelo somatório dos valores da respectiva coluna.
- Cálculo das prioridades médias locais (PML's): As PML's são as médias das linhas dos quadros normalizados.
- Cálculo das prioridades globais: Nesta etapa identifica-se um vetor de prioridades global, que armazene a prioridade associada a cada alternativa em relação ao foco principal.

3- Consistência Lógica: Segundo Saaty (2000), o ser humano tem a capacidade de estabelecer relações entre objetos ou ideias de forma que elas possam ser coerentes, de maneira que se relacionem bem entre si e suas relações apresentam consistência.

Em resumo, AHP é um método de tomada de decisão baseado em multicritério, que estão de acordo com as preferências dos tomadores de decisão.

A metodologia que será aplicada neste trabalho, utiliza como método de priorização de requisitos a técnica dólar-cem apresentadas neste capítulo.

Capítulo 3

Simulação Computacional

Como pode ser observado, um dos grandes desafios no desenvolvimento de software é compreender as necessidades dos interessados e manter as informações relativas ao software atualizadas. No entanto, implementações de melhoria de processos em empresa de software possuem alto custo ou demandam muito tempo para serem avaliados. Dentro desse contexto, o uso de ferramentas de simulação passa a ser uma alternativa para a obtenção de resultados mais rápidos e com custo reduzidos.

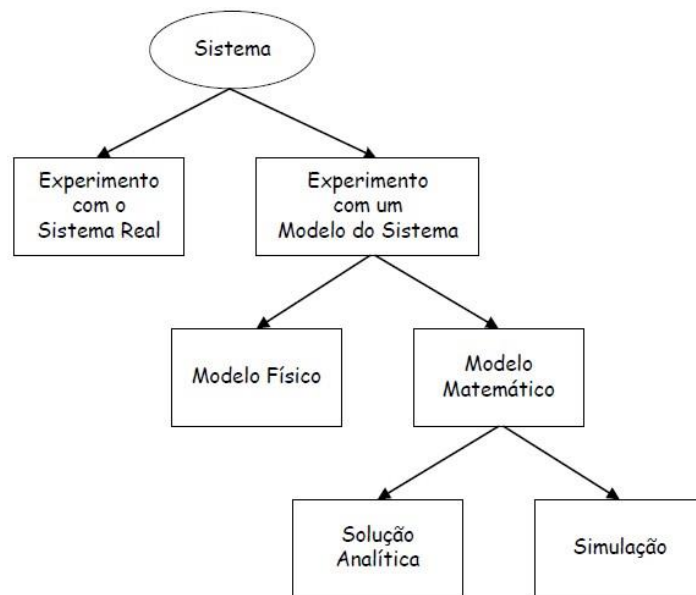
Neste capítulo são apresentados os conceitos sobre modelagem e simulação de sistemas e as vantagens e desvantagens da simulação.

3.1 Modelagem e Simulação de Sistemas

Os sistemas podem ser definidos como uma coleção de entidades, por exemplo, pessoas ou máquinas, que agem e interagem em conjunto para a realização de algum fim lógico. Na prática o significado de sistema depende dos objetivos de um estudo particular. O estado de um sistema pode ser considerado como variável, que determina o sistema em um determinado momento, em um estudo de algum banco; exemplos de variáveis de estados são: o número de clientes no banco, caixas ocupados, hora de chegada de cada cliente no banco (Law, 2007).

De acordo com a Figura 4, Leal (2011) descreve que os sistemas podem ser estudados de duas maneiras: Através da atuação direta sobre o mesmo ou pela utilização de modelos.

Figura 4 - Formas de estudar um sistema



Fonte: Law (*apud* Leal, 2011)

De acordo com a abordagem utilizada, os modelos podem ser classificados como Físicos ou Matemáticos (Leal, 2011).

- **Modelo Físico:** São baseados na construção de protótipos, que posteriormente são sujeitos às mesmas condições de funcionamento ou operação que um sistema real. Uma maquete de uma aeronave para estudar o comportamento do ar fluindo em sua fuselagem em um túnel de vento é um exemplo de aplicação de uma modelagem física (Leal, 2011).
- **Modelo Matemático:** Consiste em transformar a realidade numa relação de números, equações e/ou lógicas podendo abranger soluções analíticas ou numéricas (simulação). De acordo com Freitas Filho (2008), enquanto na abordagem analítica emprega-se o uso de um conjunto de fórmulas matemáticas que permitem calcular as respostas desejadas, a abordagem via simulação implica na utilização de um programa computacional que executa uma série de instruções que após processadas transmitem ao usuário a sensação que o modelo representa, de fato, o sistema em que está fundamentado.

Diferentes autores conceituam simulação de diferentes formas.

De acordo com Banks (1999) a simulação é uma imitação de um processo ou sistema do mundo real ao longo do tempo. Ela é usada, principalmente, para descrever e analisar o comportamento de um sistema real ou até mesmo os sistemas conceituais.

Segundo Pidd (1998), a simulação computacional consiste no uso de um modelo como base para exploração e experimentação da realidade. Ele afirma que a Simulação Computacional envolve a experimentação em um modelo desenvolvido no computador, buscando respostas (saídas do sistema) para variações nos dados utilizados e nas variáveis, visando avaliar o que aconteceria se determinada ação fosse tomada. Sendo assim, é possível observar o resultado das mudanças de diversos parâmetros, permitindo a comparação de diversos cenários.

No mundo corporativo a simulação está cada vez mais sendo aceita e utilizada. A utilização de interfaces gráficas amigáveis, aliada à geração de respostas detalhadas e mais rápidas que os métodos tradicionais, permitem que analistas de diversos segmentos passem a utilizar a simulação computacional (Gupta, 2009).

3.1.1 Componentes da Simulação

Segundo Garcia (2006) e Lamarca (2009), uma simulação pode ser vista como um conjunto de partes que se organizam e interagem para formar a estrutura do modelo. Os autores estabelecem partes destes elementos, sendo os principais definidos no Quadro 7, de acordo com a terminologia utilizada por Freitas (2008).

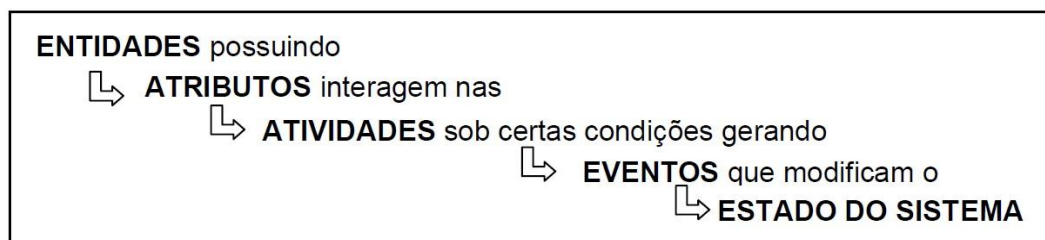
Quadro 7: Componentes relacionados a simulação e seus significados

Elementos	Definição
Variáveis de Estado	São variáveis cujos valores determinam o estado de um sistema. Constituem o conjunto de informações que permite compreender o que se está passando nele num determinado instante de tempo.
Evento	São os acontecimentos, programados ou não, que quando ocorrem alteram o estado do sistema. A chegada/saída de um cliente e o início/fim de um atendimento são exemplos de eventos.
Entidades e Atributos	Entidades são as pessoas ou objetos envolvidos no modelo estudado e podem ser dinâmicos ou estático. Cada entidade é definida por uma série de características próprias chamadas atributos. Entidades do mesmo tipo têm os mesmos atributos, embora os valores destes sejam normalmente diferentes.
Recursos e Filas de Recursos	Os recursos podem ser considerados uma entidade estática que fornece serviço às dinâmicas. Sempre que uma entidade dinâmica não puder acessar um recurso ela aguardará por ele numa fila.
Atividades	Período de tempo predeterminado que uma vez iniciado, seu final pode ser programado.

Fonte: (A autora)

A Figura 5 mostra como os componentes da simulação interagem entre si.

Figura 5: Relacionamento dos componentes da simulação



Fonte: Garcia, 2006.

3.2 Vantagens e Desvantagens do Uso da Simulação

A concorrência nas indústrias de computadores levou a avanços tecnológicos que permitem produzir produtos melhores. O hardware dos computadores tornou-se mais poderoso, mais preciso, mais rápido e prático de ser usado, o software de simulação também.

Segundo Banks (1999), o número de empresas que usam ferramentas de simulação está aumentando. Para a maioria das empresas, os benefícios do uso da simulação, está além de apenas dar uma olhada no futuro. A simulação permite testar todos os aspectos de mudanças proposto ou adição de uma nova funcionalidade sem comprometer recursos para a sua aquisição. Isso é crítico, porque uma vez que as decisões difíceis foram feitas as mudanças e as correções podem ser extremamente caras.

Outra vantagem da simulação é permitir comprimir ou expandir a simulação do tempo, é possível acelerar ou diminuir a velocidade de execução dos fenômenos para que se consiga investigar minuciosamente. Uma mudança inteira pode ser examinada em questão de minutos ou um evento de minutos podem ser examinados em horas (Banks, 1999).

Um das maiores vantagens de se usar as ferramentas de simulação é que, uma vez que um modelo válido de simulação é criado, pode-se explorar novas políticas, procedimentos operacionais ou métodos sem a despesa e a interrupção da experimentação com o sistema real. É possível observar os efeitos das mudanças no computador no lugar do sistema real (Banks, 1999).

A simulação também permite compreender melhor a interação entre variáveis que constituem sistemas muito complexos, diagnosticando problemas e obtendo informações sobre o desempenho do sistema real.

No entanto, os modelos de simulação podem ser muito complexos conforme a dificuldade da aplicação e se tornar demorado e consumir recursos elevados. Em alguns casos, os resultados apresentados pela simulação podem ser difíceis de serem interpretados, devido a simulação tentar capturar aleatoriedade de um sistema real. A simulação ainda depende da validade do modelo desenvolvido, se o modelo não representar fidedignamente o sistema ou se os dados de entrada não são confiáveis, de nada adiantará fazer um estudo detalhado dos dados de saída e encontrar uma solução para o problema (Law, 2007).

3.3 Classificação dos Modelos de Simulação

Toda simulação requer a construção de um modelo com o qual serão feitos os experimentos. Um modelo matemático estudado através da simulação é chamado de modelo de simulação (Saliby, 1989).

As classificações de modelos de simulação mais relevantes para este trabalho são descritas a seguir segundo Freitas Filho (2008):

- Estáticos ou dinâmicos: Um modelo estático visa representar um sistema em um ponto particular no tempo ou um sistema no qual o tempo não desempenha papel importante (não é levado em consideração a variável tempo. Já os modelos dinâmicos representam sistemas à medida que eles mudam ao longo do tempo.
- Determinísticos ou estocásticos: Modelo de simulação que não contém variáveis aleatórias são classificados como determinísticos, esses modelos apresentam um conjunto conhecido de entradas que resultaram em um único conjunto de saídas. O modelo de simulação estocástico tem uma ou mais variáveis aleatórias como entrada. Entradas aleatórias levam a saídas aleatórias.
- Discretos ou contínuos: No modelo discreto a passagem do tempo é feita aos saltos, entre um evento e outro, por isso é também chamada de simulação de eventos discretos (Law, 2007). Já no modelo contínuo as variáveis de estado podem variar continuamente ao longo do tempo.

Capítulo 4

Análise das Ferramentas de Simulação Computacional

Este capítulo descreve e analisa as ferramentas de simulação escolhidas, avaliando o seu apoio na tomada de decisão na priorização de requisitos. Para isso é usado um modelo que representa a aplicação da técnica \$100.

4.1 Ferramentas de Simulação

Atualmente, encontram-se disponíveis no mercado diversas ferramentas de simulação, com aplicações nos mais diversos setores da economia. O presente trabalho escolheu para sua análise comparativa 3 ferramentas de simulação, que possuem versões (com limitações) gratuitas, disponíveis para *download* e apresentam documentação para entendimento da ferramenta. São elas: Arena (*Rockwell Softwares*), ProModel (*ProModel Corporations*) e Bizagi Modeler (Bizagi) na notação BPMN (*Business Process Management Initiative*).

Arena

O Arena pertence à Rockwell Software e tem sido utilizado para simular os mais diversos ambientes (Freitas e Silva, 2007). O Arena é composto por um conjunto de blocos (ou módulos) utilizados para descrever uma aplicação real e que funcionam como comandos de uma linguagem de programação.

Os elementos básicos da modelagem em Arena são as entidades, as estações de trabalho que demonstram onde será realizado algum serviço ou transformação, e por fim, o fluxo que representa os caminhos que a entidade irá percorrer ao longo de estações.

O Arena apresenta uma interface gráfica através da qual são elaborados os modelos, auxiliando na posterior montagem de cenários e tomada de

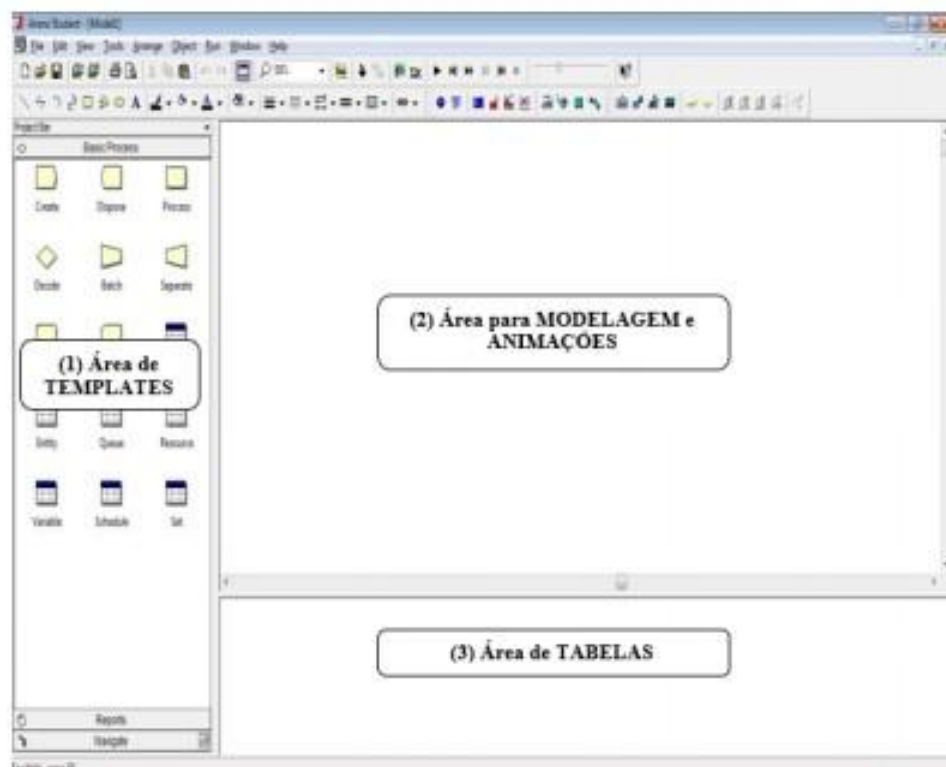
decisão. Com isso, não é necessário escrever nenhuma linha de código no software.

Segundo o Wikipedia (2016), atualmente o Arena é a ferramenta de simulação mais utilizado no mundo, com mais de 350.000 usuários e possui diversas versões.

Uma das características desta ferramenta é a possibilidade de criação do modelo computacional de forma sistêmica através de módulos, os quais permitem sua interconexão, a fim de refletir de forma global um único sistema.

O Arena apresenta uma área de trabalho objetiva, separada basicamente em três áreas, 1 Área de *Templates*, 2 Área para Modelagem e Animações e 3 Área de tabelas, conforme apresentado na Figura 6.

Figura 6: Área de trabalho – Arena



Fonte: Arena versão *Student*, 2012

O software também apresenta uma barra de menu superior com ícones intuitivos, contemplando o acesso às configurações do software, apresentado

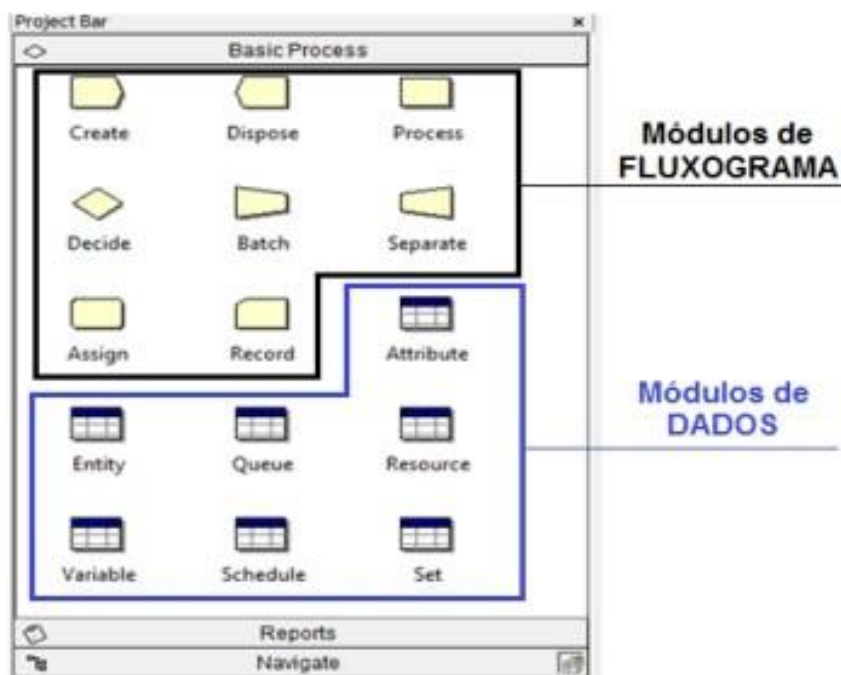
na Figura 7. Já na Figura 8 é possível visualizar o *Template Basic Process*, conjunto de módulos básicos que são utilizados para a construção do modelo.

Figura 7: Barra de comandos geral – Arena



Fonte: Arena versão *Student*, 2012

Figura 8: *Template Basic Process* – Arena



Fonte: Arena versão *Student*, 2012

Juntamente com o Arena, é fornecida uma biblioteca de pequenos modelos de simulação (*smarts*), que conseguem abordar vários aspectos de modelagem.

ProModel

A escolha para estudar essa ferramenta de simulação se deu pela sua disponibilidade e à familiarização da ferramenta por parte do autor. O ProModel foi desenvolvido pelo ProModels *Corporation*,

O Quadro 8 faz uma breve descrição dos elementos e funções básicas do ProModel segundo Barrônio (2000) e manual do software ProModel (1997).

Quadro 8: Funções básicas do ProModel

Funções	Descrição
Entidades (<i>entity</i>)	Elementos que se “movem” e sofrem processamento no modelo. Como exemplo tem-se os produtos, as pessoas, os documentos e as ordens de pedidos.
Local (<i>location</i>)	Elementos que representam lugares físicos e fixos no sistema onde ocorrem os processos. Pode ser um local, uma mesa, uma estrutura.
Chegadas (<i>arrivals</i>)	Elementos que indicam a introdução de entidades no modelo, ou seja, cada vez que uma entidade nova é introduzida no modelo uma chegada ocorre. Podem ser baseadas no tempo ou em alguma condição.
Recurso (<i>resources</i>)	Elementos de apoio a execução do processo ou de movimentação, como exemplo tem-se uma empilhadeira ou um operador.
Processos (<i>process</i>)	Define as operações realizadas sobre as entidades e a movimentação das entidades para os diferentes locais.
Rede de percurso (<i>path networks</i>)	Definem o caminho a ser percorrido pelos recursos e/ou entidades durante a a movimentação ao longo do modelo.
Turnos (<i>shifts</i>)	Representa a escala de trabalho dos recursos e locais.
Variáveis (<i>variables</i>)	São contadores que podem ser incrementados ao longo do modelo para indicação da performance do modelo.
Atributos (<i>attributes</i>)	São informações adicionais a entidades ou locais.
Matriz (<i>array</i>)	Elemento de armazenamento de dados.

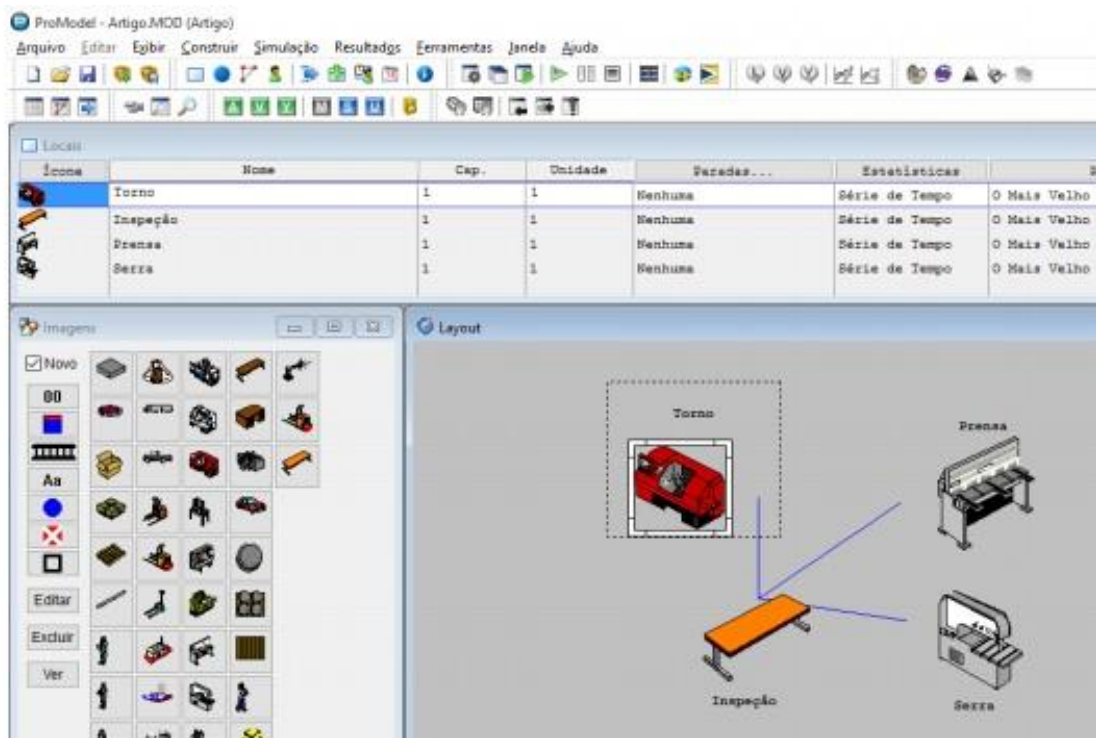
Macros (<i>macros</i>)	Permite a associação de um valor que ocorre repetidas vezes.
Sub-rotinas (<i>subroutines</i>)	Bloco lógico parametrizável que se repete em vários pontos do modelo.

Fonte (A autora)

Basicamente, a modelagem na ferramenta ProModel pode ser realizada por meio de ícones e imagens, sendo o modelo computacional obtido a partir de uma construção gráfica.

Como exemplificação, a Figura 9 apresenta a aba Local, já mencionada no quadro acima.

Figura 9: Área de Trabalho – ProModel



Fonte: ProModel versão *RunTimeSilver*, 2011

A Figura 10 representa a barra de menu superior, onde se encontra o principal ponto de praticidade do sistema. Para a configuração básica de um modelo, é necessário acessar minimamente quatro abas do sistema, por meio dos botões disposto na barra de menu superior: 1 Locais, 2 Entidades, 3

Processos e 4 Chegadas, conforme apresentado na Figura 11. A ferramenta permite a configuração dos parâmetros no idioma português.

Figura 10: Barra de comandos geral – ProModel



Fonte: ProModel versão *RunTimeSilver*, 2011

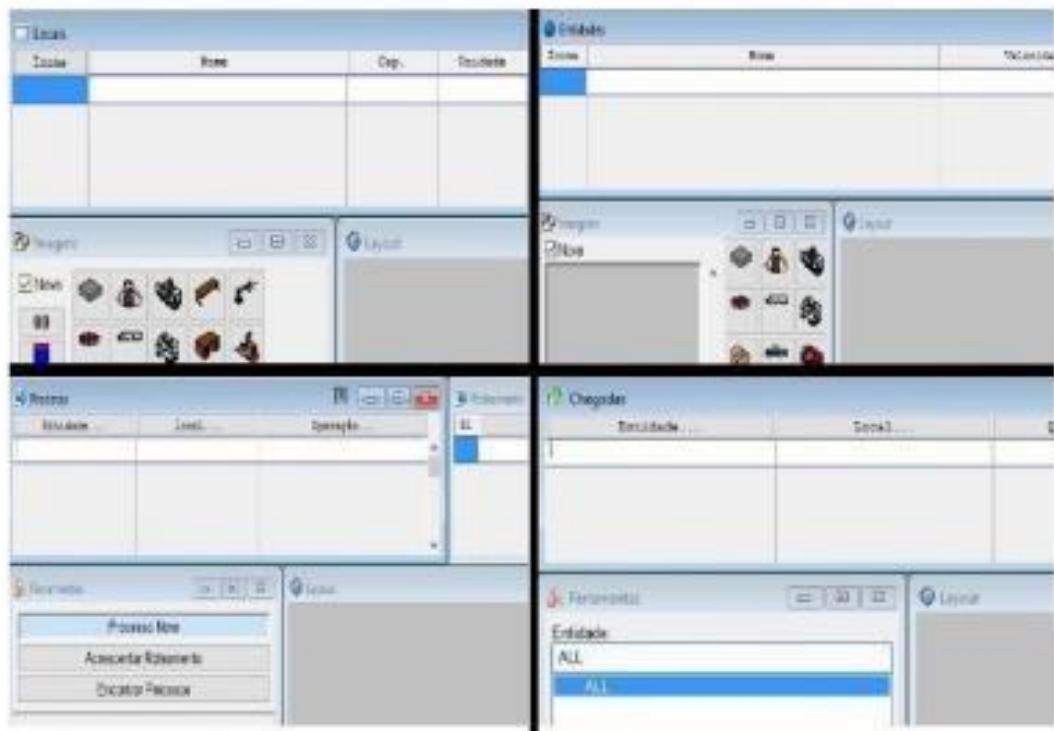
Figura 11: Barra de comandos básicos –ProModel



Fonte: ProModel versão *RunTimeSilver*, 2011

A ferramenta de simulação ProModel demanda atenção do usuário em relação às suas abas, por possuírem grande similaridade entre si, conforme demonstrado na Figura 12.

Figura 12: Áreas para modelagem – ProModel



Fonte: ProModel versão *RunTimeSilver*, 2011

Bizagi Modeler na notação BPMN (Business Process Model and Notation)

O BPMN é uma notação gráfica que captura a lógica das atividades e todas as informações necessárias para análise, simulação e implementação de um processo (Vieira, 2014). Possui uma linguagem gráfica de fácil entendimento e visualização. Os principais objetos são descritos no Quadro 9.

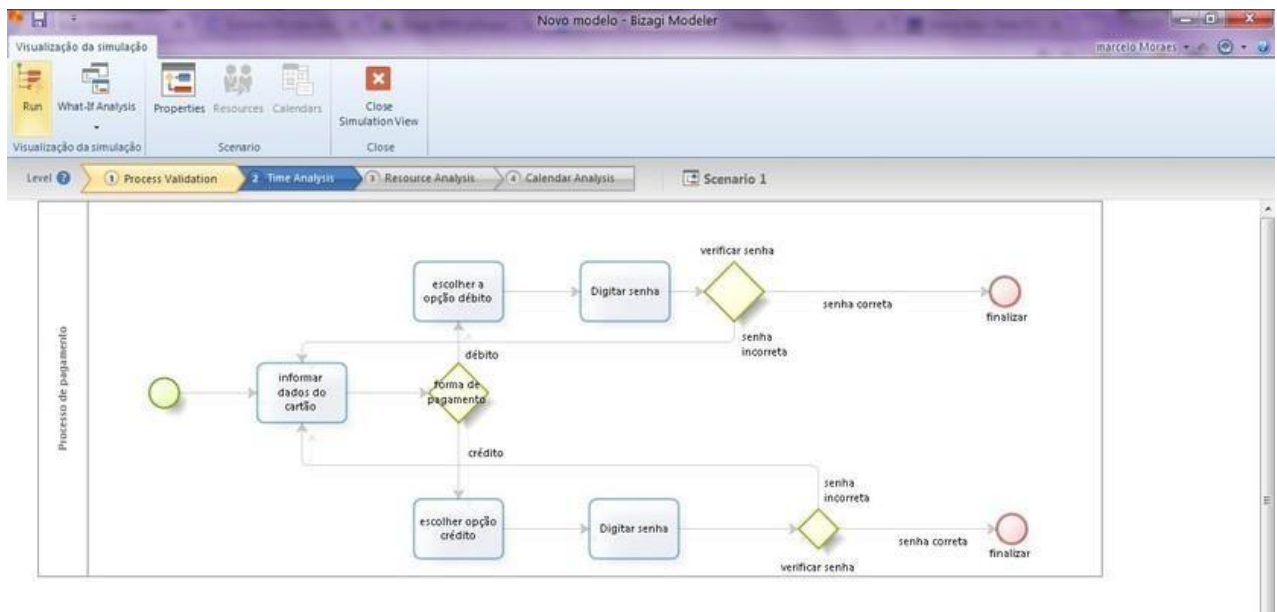
Quadro 9: Componentes da notação BPMN

Objetos	Descrição
Piscina e raias	Piscina é um espaço onde irá conter os passos de um único processo, seu nome irá representar o nome do processo. As raias são as divisões da piscina que representa setores.
Eventos	Eventos representam acontecimentos que podem acontecer durante o curso de um processo e afeta seu fluxo. Podem ser do tipo evento de início, intermediário e fim, podendo existir eventos dos tipos de especificações.
Atividades	Tarefa representa uma atividade simples usada quando um trabalho não pode ser dividido em mais detalhes. Existem vários tipos de tarefas, dentre elas: Serviço (quando a tarefa é realizada pelo sistema), recebida, envio e manual.
Subprocesso	Atividade composta cujos detalhes são definidos em um novo fluxo de atividades.
Gateways	São elementos usados para controlar as convergências e divergências do fluxo.
Indicativos de fluxo	Indicam o caminho seguido pelo processo, troca de mensagem entre dois processos e associação.
Artefatos	Servem para anotação, agrupamento, objeto de dados e depósito de dados.

Fonte (A autora)

A simulação na ferramenta Bizagi Modeler permite o aprimoramento dos modelos construídos na notação BPMN e suporta rigorosos métodos de análises. Fornece ambiente gráfico para definição de regras de negócio que serão seguidas durante a execução do processo, permite o acompanhamento de cada passo do processo, além de fornecer relatórios para monitoramento do negócio em tempo real (Bizagi, 2016). A Figura 13 demonstra a área de simulação da ferramenta de simulação Bizagi Modeler

Figura 13: Área de simulação – Bizagi Modeler



Fonte: Bizagi versão 3.1, 2016

4.2 Simulação Utilizando as Ferramentas

Através das ferramentas de simulação descritas teve-se o objetivo de aplicar a técnica de priorização de requisitos \$100. Para isso foram determinados os *stakeholders*, requisitos, quantidade de requisitos a serem priorizados, tempo para montar o *ranking* de requisitos, monitorar a simulação em relação aos requisitos que já foram priorizados, início da priorização, final da priorização e tempo total gasto para a priorização.

O sistema simulado em cada uma das ferramentas foi o processo de aplicação da técnica \$100, baseado nos modelos e simulações realizadas por Timóteo, 2016. Com relação às variáveis de resposta, que foram os critérios analisados, considerou-se a avaliação do tempo total gasto na priorização. Conforme visto anteriormente, nesta técnica são fornecidas 100 unidades imaginárias aos *stakeholders* e cada um fica responsável por distribuir os valores entre os requisitos a serem priorizados. Em seguida, é feita a ordenação dos requisitos.

Para a concepção dos cenários, foram levados em consideração: a quantidade de *stakeholders* presentes, quantidade de requisitos que chegam para serem priorizados, tempo necessário para que seja executada cada atividade. Para estimar o tempo das atividades (processos), foi realizada uma conversa com um engenheiro de requisitos de uma empresa do Porto Digital, para saber quantos requisitos em média chegam por semana para serem priorizados e quanto tempo era necessário para que fosse feita as etapas da priorização.

Com relação aos cenários criados, considerou-se uma quantidade fixa de requisitos de entrada = 5 requisitos., e a seguinte variação:

- Cenário 1: Está presente 1 *stakeholder*, são necessários 5 minutos para analisar cada requisito, 24 minutos para distribuição dos dólares e o prazo para priorização total são de 2 horas.
- Cenário 2: Estão presentes 2 *stakeholders*, o primeiro leva 5 minutos para analisar cada requisito, enquanto o segundo é um pouco mais lento, sendo necessário 6 minutos para análise de cada requisito; ambos demoram 20 minutos para realizar a distribuição dos dólares entre os requisitos e o prazo para entregar a lista de requisitos com ordem de prioridades é de 2 horas.

A diferença entre o primeiro e o segundo cenário, está em relação à quantidade de *stakeholders* que participam; e o seu impacto no tempo. No cenário 1 temos a participação de um único *stakeholders* com o tempo gasto em minutos de 5, já no segundo cenário temos a participação de 2 *stakeholders* sendo o segundo com tempo de 6 minutos para analisar cada requisito.

Os critérios usados para a comparação das ferramentas foram:

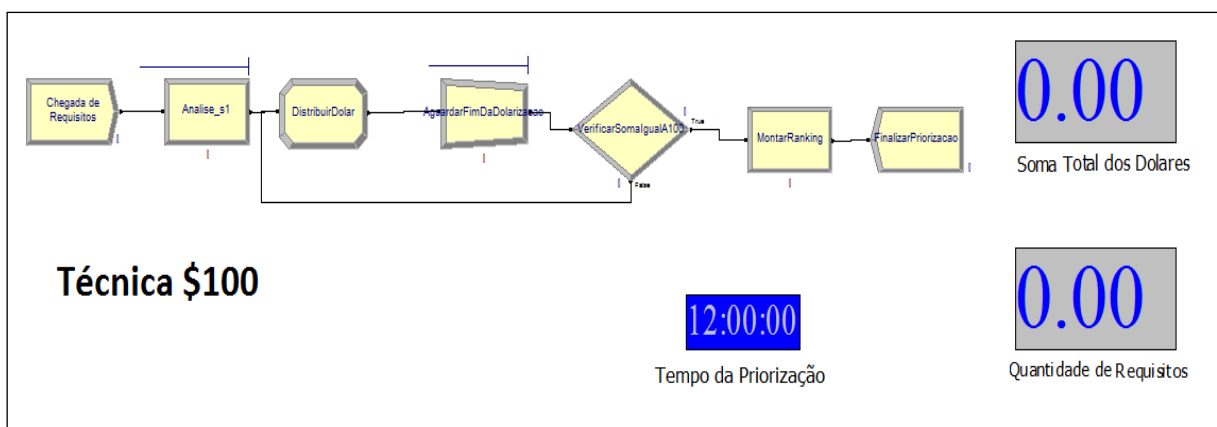
- 1- Curva de aprendizagem do software;
- 2- Facilidade de uso, como: Usabilidade do software e etapas de configuração;

- 3- Necessidade de conhecimento em programação, pois boa parte dos requisitos são definidos por gerentes de projetos, pelos quais não possuem formação em tecnologia da informação;
- 4- Interface gráfica, que facilite o entendimento e a visualização dos requisitos e de todo o processo;
- 5- Documentação para entendimento da ferramenta;
- 6- Conceitualizar requisitos;
- 7- Conceitualizar *stakeholder*;
- 8- Determinar variáveis que sofrem alterações durante a simulação, por exemplo, soma total dos dólares devem ser igual a 100 ao final da simulação.
- 9- Tempo total da simulação referente ao processo de priorização de requisitos.

4.2.1. Aplicando a técnica \$100 na ferramenta Arena

A Figura 14 mostra o modelo adotado para simular no Arena a técnica de priorização de requisitos \$100. Nela é utilizado o cenário 1, em que se considera a existência de um único *stakeholder* para priorizar os requisitos.







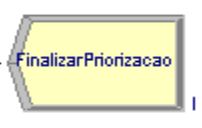



Figura 14: Técnica \$100 – Arena



Fonte: (A autora)

O Quadro 8 apresenta descrição dos blocos utilizados para modelar da técnica \$100 no software Arena.

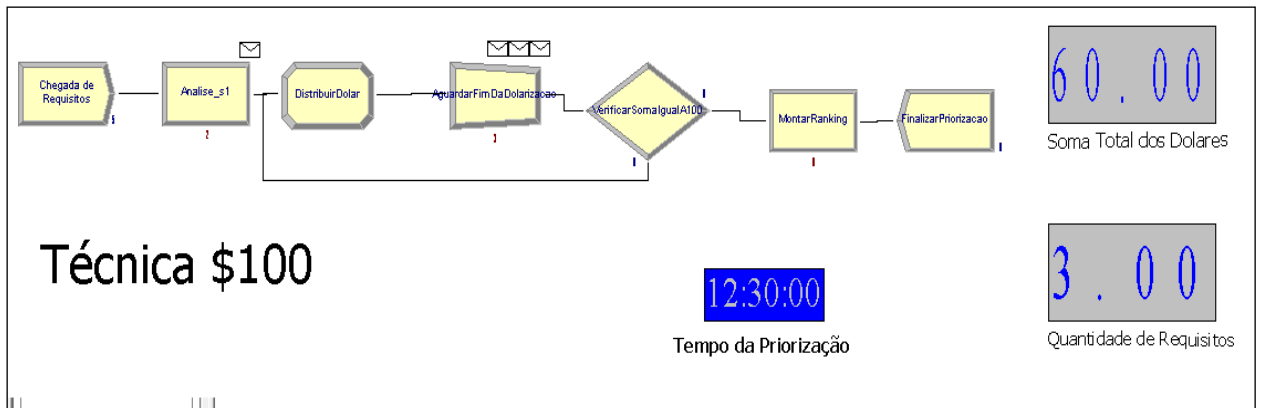
Quadro 8: Blocos utilizados para simulação – software Arena

Bloco de Fluxo	Descrição	Configuração
	Bloco responsável pela criação das entidades (requisitos) que fluirão pelo sistema.	<i>Entity type = Requisito; Entities per arrival = 5; Max Arrivals = 1.</i>
	Processo no qual o <i>stakeholder</i> conhece e compreende os requisitos.	<i>Resource: Stake 01; Value= 5 min.</i>
	Bloco onde são distribuídos os dólares entre os requisitos.	<i>Entity type = Requisito; SomaTotal = SomaTotal + Requisito; QuantidadeRequisito= QuantidadeRequisito + 1;</i>
	Agrupa os requisitos até que todos tenham recebido algum valor	<i>Batch Size= 5; Entity Type = Requisito; 2.5.4 Rule = Entity.Serial Numbers.</i>
	Bloco de Decisão que verifica se a soma dos dólares atribuídos ao conjunto de requisitos totaliza um valor menor ou igual a 100.	<i>Type = Condition; Variable = SomaTotal;</i>
	Organiza os requisitos por ordem de priorização. Este processo passa um atraso constante de tempo por requisito.	<i>Delay= 20 min.</i>
	Processo que representa a finalização do processo de priorização.	
Bloco de Dados	Descrição	Configuração
	Bloco onde é realizada a configuração das entidades criadas.	<i>Type= Requisitos.</i>
	Bloco onde é realizada a configuração dos recursos criados.	<i>Name= Stake01;</i>
	Bloco onde é realizada a configuração das variáveis que farão parte do processo.	<i>QuantidadeRequisito = 5</i>

Fonte: (A autora)

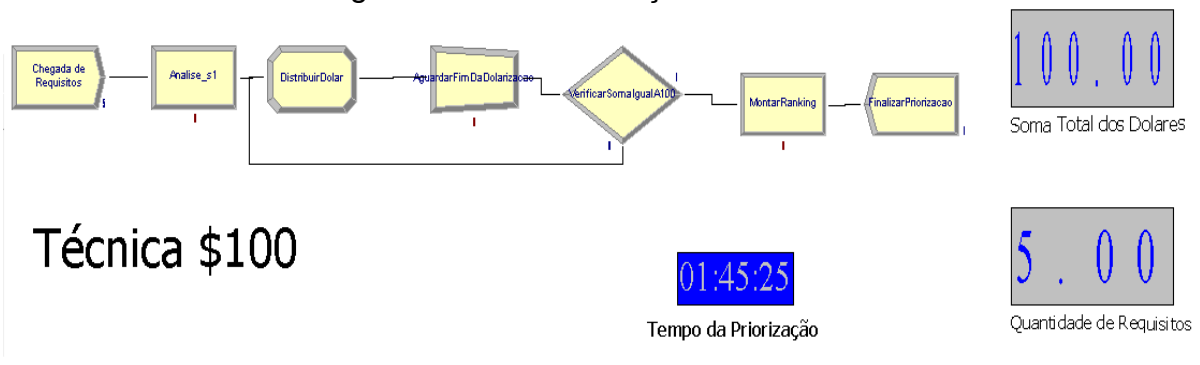
A Figura 15 demonstra a execução do processo da simulação e a Figura 16 o final da simulação e os respectivos valores obtidos.

Figura 15: Durante processo de simulação – Arena



Fonte: (A autora)

Figura 16: Final simulação – Arena



Fonte: (A autora)

O resultado após a execução do modelo no Arena para o cenário 1, em que a priorização é realizada por um único *stakeholder* é apresentado na Tabela 2.

Tabela 2: Relatório de simulação do cenário 1 no Arena

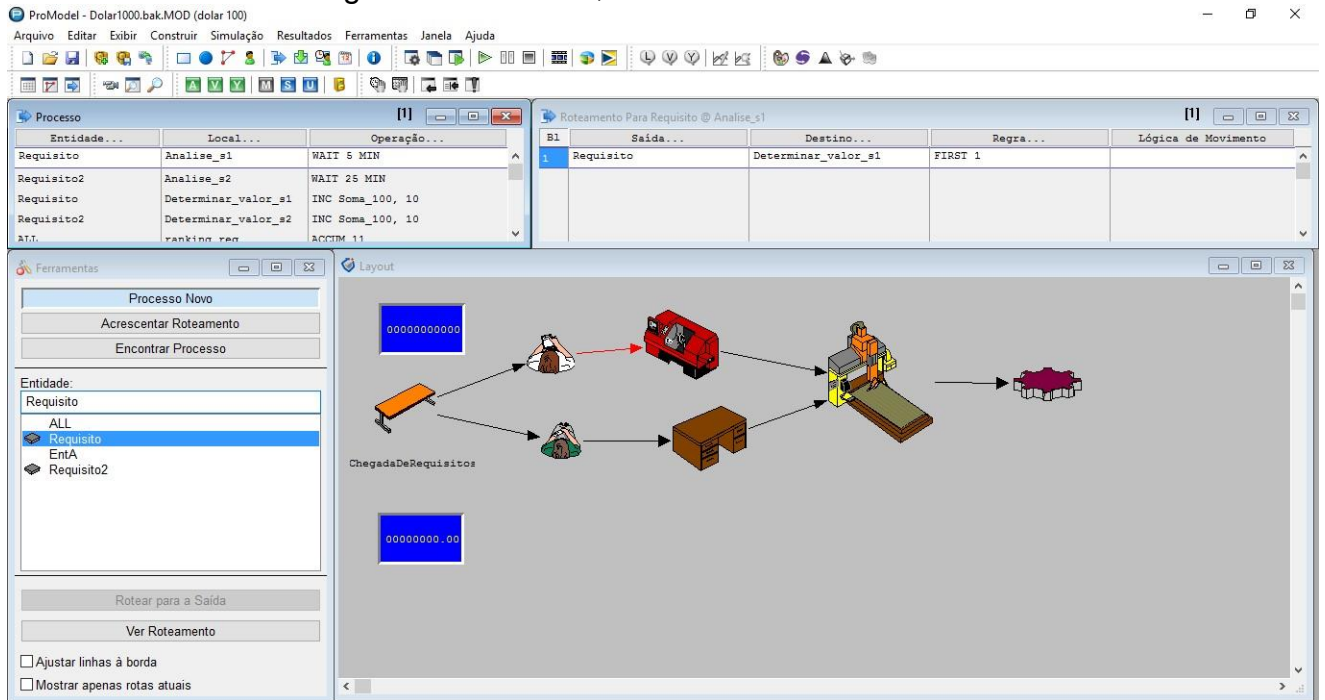
Cenário 1	Tempo para Conhecer Requisitos	Tempo para Distribuir os Dólares	Tempo para Montar Ranking	Requisitos. Priorizados	Soma Total dos Dólares	Tempo Total ao final da simulação do processo de priorização de requisitos
Stake 01	25 minutos	60 minutos	20 minutos	5	100 dólares	105 minutos

Fonte: (A autora)

4.2.2 Aplicando a técnica \$100 na ferramenta ProModel

Adotando o segundo cenário foi possível obter o modelo demonstrado na Figura 17 para a ferramenta ProModel, estão representados os locais onde para cada local alguma ação pode ser tomada.

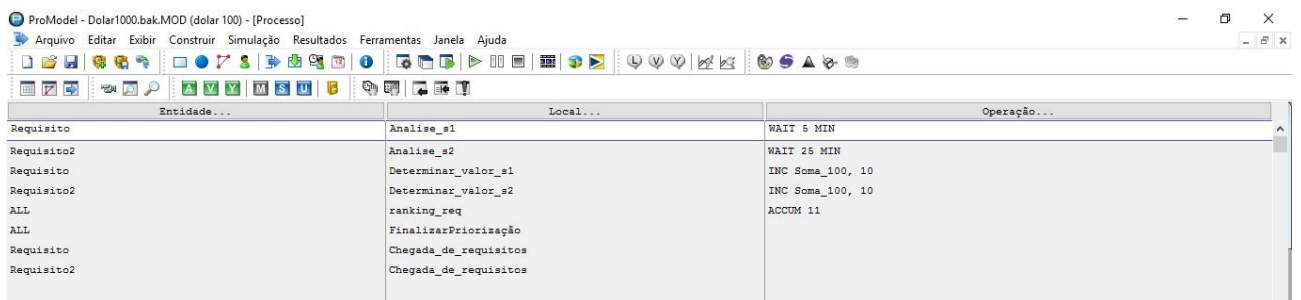
Figura 17: Técnica \$100 – ProModel



Fonte: (A autora)

Os processos atribuídos aos seus locais e os valores de operações necessários para se realizar a priorização de requisitos, utilizando a técnica \$100, são demonstrados na Figura 18.

Figura 18: Recursos utilizados – ProModel



Fonte: (A autora)

O resultado após a execução do modelo no ProModel para o cenário 2, em que a priorização é realizada por dois *stakeholders* e cada um possui um tempo para analisar os requisitos, é apresentado na Tabela 3.

Tabela 3: Relatório de simulação do cenário 2 no ProModel

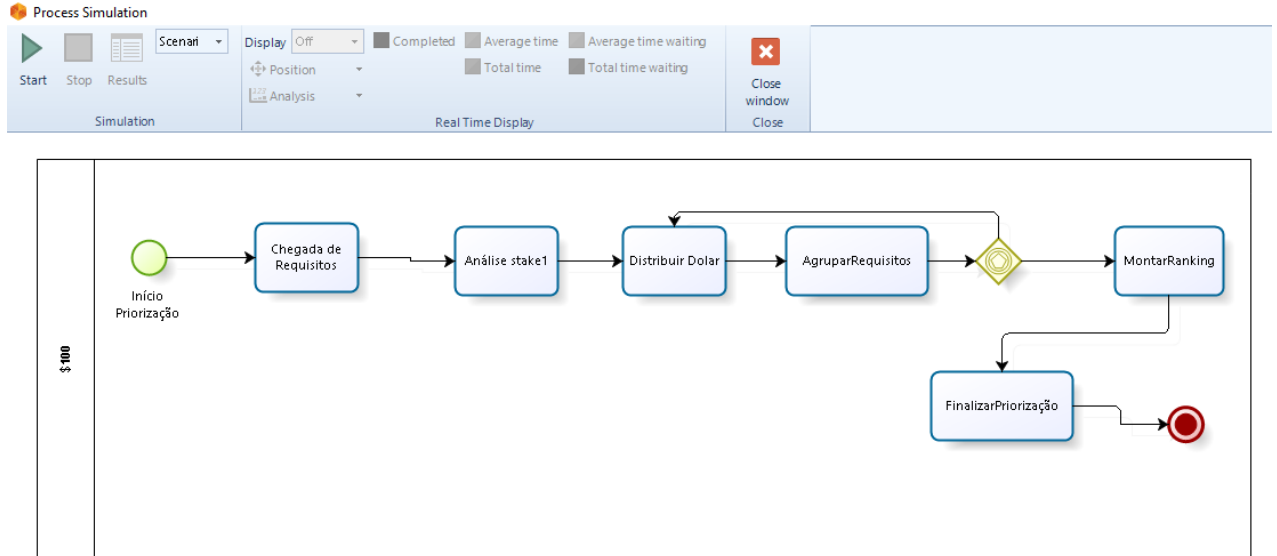
Cenário 1	Tempo para Conhecer Requisitos	Tempo para Distribuir os Dólares	Tempo para Montar Ranking	Tempo Total	Requisitos priorizados	Soma Total dos Dólares	Tempo Total da Simulação
Stake 01	25 minutos	60 minutos	20 minutos	105 minutos	5	100	-
Stake 02	30 minutos	60 minutos	20 minutos	110 minutos	5	100	-
-	-	-	-	-	-	-	110minutos

Fonte: (A autora)

4.2.3 Aplicando a técnica \$100 na ferramenta Bizagi Modeler na notação BPMN

Partindo do primeiro cenário em que apenas 1 *stakeholder* fará a priorização, sendo ele o recurso utilizado, foi possível apenas a criação de recursos e entidades na ferramenta Bizagi Modeler. A Figura 19 demonstra como ficou o modelo adotado para a priorização de requisitos, adotando a técnica \$100.

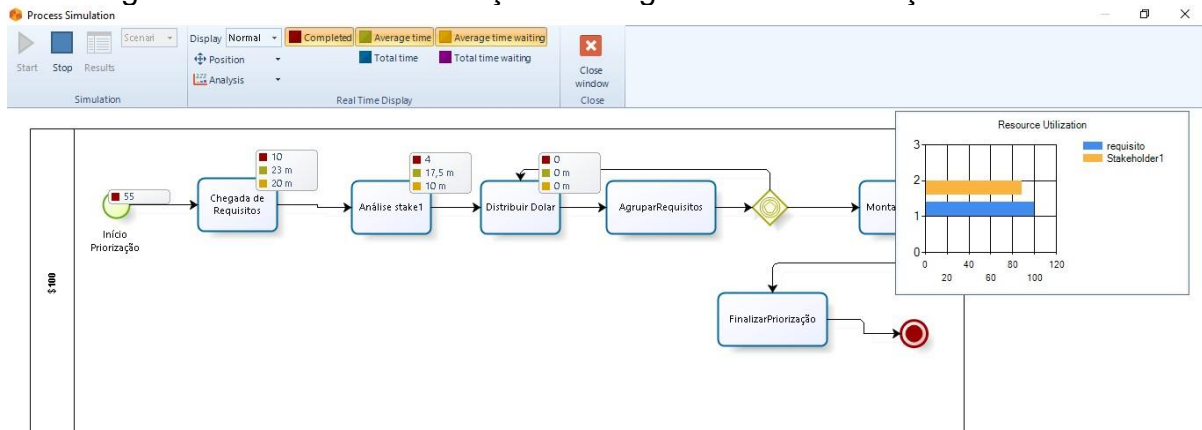
Figura 19: Técnica \$100 – ferramenta Bizagi Modeler na notação BPMN



Fonte: (A autora)

Durante a simulação foi possível observar a alocação dos recursos e o estado de cada bloco conforme demonstrado na Figura 20.

Figura 20: Durante a simulação – Bizagi Modeler na notação BPMN



Fonte: (A autora)

O resultado após a execução do modelo no Bizagi Modeler para o cenário 1, em que a priorização é realizada por um único *stakeholder* é apresentado na Tabela 4.

Tabela 4: Relatório de simulação do cenário 1 no Bizagi Modeler

Cenário 1	Tempo para Conhecer Requisitos	Tempo para Distribuir os Dólares	Tempo para Montar Ranking	Requisitos Priorizados	Soma Total dos Dólares	Tempo Total ao final da simulação do processo de priorização de requisitos
Stake 01	25 minutos	60 minutos	20 minutos	5	Não foi possível determinar	105 minutos

Fonte: (A autora)

4.3 Análise das Ferramentas e Considerações

A análise das ferramentas de simulação foi realizada seguindo os critérios utilizados para comparar as ferramentas (descritos na seção 4.2 deste trabalho), e pela execução de experimentos considerando o processo de priorização de requisitos utilizando a técnica \$100.

Para construir o modelo do cenário utilizado na ferramenta Arena, foi necessário um tempo considerável, uma vez que a curva de aprendizagem do Arena é longa. Não foi necessário ter conhecimento de programação para construção do modelo. Um problema encontrado foi a falta de ilustração de um ambiente real, uma vez que a interação com o ambiente do projeto não se assemelha a uma situação real. Apresentou documentação de apoio para criação do modelo, foi possível determinar requisitos, *stakeholders*, dados de saídas que auxiliaram na tomada de decisão.

A flexibilidade da ferramenta de simulação ProModel permitiu modelar um sistema que atendesse ao cenário da priorização. Foi necessário também um tempo considerável para criação do modelo, pois sua curva de

aprendizagem também é longa. Foi preciso ter um pouco de conhecimento em programação para criação dos processos em relação aos comandos utilizados na definição das operações, o *Logic Builder*, em que é possível validar se o modelo criado não possui nenhum erro, determinando o que será feito no processo. Um problema encontrado no software foi a interface gráfica que fugiu muito do contexto utilizado. O software disponibiliza uma boa documentação para criação de um modelo e em português. Foi possível criar entidade do tipo requisito, conceitualizar *stakeholders* e determinar dados de saída que auxiliaram na tomada de decisão.

Já na terceira ferramenta de simulação utilizada o Bizagi Modeler com notação BPMN, não foi necessário muito tempo para criação do modelo, uma vez que é um software simples e de fácil entendimento. Seu principal objetivo é a modelagem do gerenciamento de processos de negócio, o que dificultou na criação de elementos fundamentais para priorização, como por exemplo, dados de saída para a tomada de decisão. Não foi possível conceitualizar requisitos nem atribuir valores aos mesmo, o que dificultou em um modelo válido para a priorização de requisitos baseada na técnica \$100. Sua interface gráfica também não atendeu a ilustração de um ambiente real. Não foi necessário conhecimento em tecnologia da informação para a construção do modelo.

Após esta análise podemos considerar que as ferramentas de simulação Arena e ProModel atenderam o domínio da priorização de requisitos, uma vez que foi possível conceitualizar sobre requisitos e *stakeholders*, criar variáveis que determinaram dados de saída que auxiliaram na tomada de decisão, como a soma total dos dólares e tempo total da simulação, criar modelos que fossem válidos e atendessem aos cenários e boa parte dos critérios mencionados.

Capítulo 5

Considerações Finais

5.1 Conclusão

A elicitação e priorização de requisitos são etapas muito importantes na busca pelo sucesso de projetos de desenvolvimento de software, pois, por meio destas atividades todas as demais atividades do projeto são direcionadas. A existência de diversas técnicas de priorização faz surgir a necessidade de definir qual a melhor técnica para ser aplicada no desenvolvimento do projeto.

Neste trabalho, foi demonstrado, tanto pela pesquisa na literatura Gupta (2009) e Freitas Filho (2008), quanto nos experimentos realizados, que é possível através da técnica de simulação, modelar, experimentar, analisar, verificar e concluir a respeito de determinado processo, sem a necessitar de implantá-lo. Banks (1999), afirma que uma das maiores vantagens do estudo e análise da simulação, é que, uma vez desenvolvido um modelo de simulação válido, pode-se incorporar modificações sem se preocupar com custos efetivos ou perturbação do sistema real. Diretamente relacionada a eficácia da modelagem e da análise está a ferramenta de simulação (Gupta, 2009).

Diante deste contexto, foram estudadas 3 ferramentas de simulação, com o objetivo de determinar qual a que melhor se encaixa no domínio da priorização de requisitos. Apenas o Arena e ProModel possuem recursos de modelagem suficientes para tomada de decisão no domínio da priorização. Apesar da aplicação inicial do ProModel ter sido voltada para a manufatura, sua flexibilidade permite a modelagem de diferentes sistemas. Muitos recursos de suporte à modelagem, oferecidas pelas ferramentas de simulação escolhidas, são similares em aplicação e bastante integradas aos softwares, o que facilitou o processo de mapeamento do modelo. Já a simulação pela ferramenta Bizagi Modeler utilizando a notação BPMN apresentou bastantes restrições, como por exemplo, não conseguir determinar a soma total dos

dólares, não sendo possível chegar a uma conclusão final para o processo da priorização requisitos utilizando a técnica \$100 escolhida para este trabalho.

O tempo de construção do modelo para as ferramentas de simulação Arena e ProModel pode ser considerado similar e alto, visto que, a curva de aprendizagem das ferramentas é longa. Um problema encontrado nas três ferramentas de simulação utilizadas neste trabalho é a falta de ilustração de um ambiente real da atividade que está sendo simulada, a interação com o ambiente do projeto não se assemelha com o mundo real. No Arena os módulos considerados básicos já foram suficientes para as necessidades do modelo construído. No ProModel, o que facilitou na hora da aplicação da técnica \$100, mas que é necessário um pouco de conhecimento de programação, foram os comandos utilizados na definição das operações, o *Logic Builder*.

5.2 Trabalhos Futuros

Como proposta de trabalhos futuros propõe-se:

- Analisar outras ferramentas de simulação e outras técnicas de priorização de requisito no mesmo contexto.
- Implementar uma ferramenta de simulação específica para priorização de requisitos;
- Capturar dados reais para melhor criação de cenários;
- Criar uma ferramenta multiplataforma para auxiliar no processo de priorizar requisitos

Bibliografia

- [1] Achimugu P.; Selamat, A.; Ibrahim, R.; Mahrin, M. N. A **Systematic Literature Review of Software Requirements Prioritization Research**, [Journal], Disponível: www.elsevier.com/locate/infsof, fev, 2014.
- [2] Allen, J.H., *et al.* **Software Security Engineering: A Guide for Project Managers**. Addison Wesley Professional, 2008. 368p.
- [3] Arena. *Rockwell Automation. Version 14.0000 Student. United States of America*. 2012.
- [4] Azar, J.; Smith, R. **Value-Oriented Requirements Prioritization in a Small Development Organization**. IEEE Software, 2007.
- [5] Banks, J. **Introduction to Simulation**. In Proceedings of the Winter Simulation Conference, 1999.
- [6] Barrônio, S., **Desenvolvimento de modelos de simulação computacional para análise e melhorias de sistemas produtivos**. Dissertação de Mestrado, Porto Alegre: Programa de Pós- Graduação em Engenharia de Produção/ Universidade Federal do Rio Grande do Sul, 2000.
- [7] Berander, P. **Prioritization of Stakeholder Needs in Software Engineering - Understanding and Evaluation**. Ronneby, Sweden, pp 172 Licentiate Series No 2004:12 - Blekinge Institute of Technology, School of Engineering, Department of Systems and Software Engineering, Sweden, 2004.
- [8] Berander, P., & Andrews, A., **Requirements prioritization**. In Engineering and managing software requirements (pp. 69-94). Springer Berlin Heidelberg, 2005.

- [9] Bizagi. **Bizagi. 2016**. Disponível em:<<http://www.bizagi.com>> . Acesso em: 30 janeiro, 2016.
- [10] Booch, G.; Rumbaugh, J.; Jacobson, I. **UML, Guia do Usuário**. Rio de Janeiro: Campus. 472p. 2000.
- [11] Carlshamre, P. A **Usability Perspective on Requirements Engineering – From Methodology to Product Development**. Dissertation No. 726 – Linköpings Universitet, Linköping Studies in Science and Technology, Department of Computer and Information Science, Sweden, 2001.
- [12] Cordeiro, A. G.; **Priorização de requisitos e avaliação da qualidade de software segundo a percepção dos usuários**. 2010. 97 f. Dissertação(Engenharia de Produção) - Universidade Estadual Norte Fluminense,Campos dos Goytacazes, 2010.
- [13] Cordeiro, A. G.; Freitas, A.L.P. **O cenário atual da qualidade de software**. Anais do XV Simpósio de Engenharia de Produção, Bauru – SP. 2008.
- [14] Cordeiro, A. G.; Moll, R. N. **Pesquisa de Satisfação de Usuários de Software de Gestão Hospitalar Utilizando os Critérios da ISO 9126**. 2006.
- [15] Freitas Filho, P.J, **Introdução à Modelagem e Simulação de Sistemas com Aplicações em Arena**. 2ed.- Florianópolis: Visual Book, 2008.
- [16] Freitas e Silva, Liane Márcia; Pinto, Marcel de Gois; Subramanian, Anand. **Utilizando o software Arena como ferramenta de apoio ao ensino em Engenharia de Produção**. XXVIII Encontro Nacional de Engenharia de Produção. Foz do Iguaçu, 2007.
- [17] Garcia, Leonardo Claro. **Dimensionamento de recursos de atendimento móvel de urgência da região metropolitana II do estado do Rio de Janeiro**. Dissertação (Mestrado em Engenharia de Produção). Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2006.

- [18] Gupta, A., Verma, R., & Singh, K.; **Smart Sim selector: a software for simulation software selection**. International Journal of Engineering (IJE), 3(3), 175, 2009.
- [19] Jair C. L. **O Processo de Desenvolvimento de Software**, 2000 Disponível em: < <https://www.dimap.ufrn.br/~jair/ES/c2.html> >. Acesso em 23 de maio 2017.
- [20] Júnior, Jose Maurício Silva, **G-4REPrioritization: Um Guia para Apoio à Escolha de Técnicas de Priorização de Requisitos**. Dissertação de Mestrado, Recife: Programa de Pós- Graduação em Engenharia da Computação/ Universidade de Pernambuco, 2015.
- [21] Karlsson, J.; Ryan, K. Supporting the selection of Software Requirements. In:8th **International Workshop on Software Specification and Design**, 1996. *Proceedings of the 8th International Workshop on Software Specification and Design (IWSSD '96)*, 1996, p. 146-149.
- [22] K. Pohl, C. Rupp, **Fundamentos da Engenharia de Requisitos**, 2010.
- [23] Lamarca, Braz Rafael da Costa. **Utilização da simulação para avaliação de equipes cirúrgicas de revascularização miocárdica em relação ao Australian National Diagnosis Related Group (AN-DRG) em um hospital privado**. Dissertação (Mestrado em Engenharia de Produção). Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2008.
- [24] Lamounier H. Disponível em:< <http://www.devmedia.com.br/atividades-basicas-ao-processo-de-desenvolvimento-de-software/5413>>. Acesso em 23 de maio de 2017.
- [25] Larman, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos e ao Processo Unificado**. 2. ed., Porto Alegre: Bookman, 2004.
- [26] Law, A.: **Simulation Modeling & Analysis**, 4th ed., McGraw-Hill, p.70, 2007.

- [27] Leal, Leonardo Rosas. **Simulação de eventos discretos aplicada ao gerenciamento de prazo em projetos: um estudo de caso de projeto logístico na indústria de Óleo & Gás.**Dissertação (Mestrado em Engenharia de Produção). Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2011.
- [28] Leffingwell, D. **Agile Software Requirements**. 1st edition. Addison-Wesley,2011. Kindle Edtion.
- [29] Leffingwell, D.; WIDRIG, D.; **Managing Software Requirements**. Addison Wesley, Second Edition, 2003.
- [30] Lehtola, L.; Kauppinen, M. **Empirical evaluation of two requirements prioritization methods in product development projects**. *In Software Process Improvement* ,p. 161-170 , Springer Berlin Heidelberg, 2004.
- [31] Miranda, E Time Boxing Planning: bumered Moscow Rules. Institute for Sonnare Research. Carnegie Mellon University. Sep 2011 Disponível em: http://msei.isri.cmu.edu/publications/miranda_mirandahumeredmuscuwrules.pdr> Acesso em 25 der 2013 odic.
- [32] Moreira, C. **Modelos para gestão de projetos na engenharia de software**. Palhoça, SC.(2011).
- [33] Nelson Sakurada: **Aplicação de Simuladores de Eventos Discretos no Processo de Modelagem de Sistemas de Operações de Serviços**, 2009. Disponível em :< http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2009000100004 >. Acesso em 2 de maio 2017.
- [34] Oliveira, R. R.: A Técnica de Priorizaçãp MoSCoW, 2014. Disponível em :< https://issuu.com/ronielton/docs/artigoprince2moscow2014_mpbr > . Acesso em 23 de maio 2017.
- [35] Ogc. **Gerenciando Projetos de Sucesso** com PRINCE2™. Norwich, Norfolk, England: Office of Government Commerce: The Stationery Office (TSO), 2011.

- [36] Pedgen, C. D., Shannon, r.e., Sadowski, r.p. **Introduction to Simulation Using SIMAN**. 2 ed. McGraw-Hill, Inc. New York, 1995.
- [37] Pidd, M., **Modelagem empresarial: ferramentas para tomada de decisão**. Porto Alegre: Artes Médicas, 1998.
- [38] Plínio, V.: **Conteúdo de Qualidade Sobre Engenharia de Software**, 2015. Disponível em :< <http://www.ateomomento.com.br/priorizacao-de-requisitos>>. Acesso em 10 de abril 2017.
- [39] Pohl, Klaus., **Requirements Engineering: Fundamentals, Principles, and Techniques**, 2010 | ISBN-10: 3642125778.
- [40] Pressman, R. S. **Software Engineering, a Practitioner's Approach – Alternate Edition**. Seventh Edition. McGraw Hill, 2010.
- [41] ProModel. *Promodel Corporation. Version RunTimeSilver 2011. United States of America*. 2011.
- [42] Qiao, M. **The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review**, [Dissertação], Auckland University of Technology, 2009.
- [43] Saaty, T. L, **The Analytic Hierarchy Process**. New York: McGraw-Hill International, 1980.
- [44] Saliby, E. **Repensando a Simulação: a amostragem descritiva**. São Paulo: Atlas, 1989
- [45] Sommerville, I. ; Sawyer, P. **Requirements Engineering: A Good Practice Guide**. 1. ed. New York: John Wiley & Sons, 1997.
- [46] Sommerville, I. **Engenharia de software**. São Paulo. 6. ed. Pearson Education Companion, 2003.
- [47] Sommerville, I. **Engenharia de Software**. 8. ed., São Paulo: Pearson Addison Wesley, 2007.
- [48] Thayer, M. Dorfman. **Software Requirements Engineering**. IEEE Computer Society Press, 1993.

- [49] Timóteo, S., **Conceitualização de uma Ferramenta de Simulação de Domínio Específico para Priorização de Requisitos**. Dissertação de Mestrado, Recife: Programa de Pós- Graduação em Engenharia da Computação/ Universidade de Pernambuco, 2016.
- [50] Turley, F. **Manual de Treinamento do PRINCE2® Foundation**. Tradução de Ernani Marques da Silva; Amintas Maciel Teixeira. São Paulo: Management Plaza International, v. 1.15.br, 2012.
- [51] Vestola, M. A **Comparison of nine basic techniques for requirements prioritization**. Helsinki University of Technology, 2011.
- [52] Vieira, B. B. **Curso de introdução à diagramação de processos com o software Bizagi Process Modeler**, 2014
- [53] Weber, S. ; Hauck, J. ; Wangenheim, C. **Estabelecendo processos de software em micro e pequenas empresas**. SBQS - Simpósio Brasileiro de qualidade de software, Porto Alegre, 2005.