

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

JULIANO DERTZBACHER

**Um Modelo de Simulador para Ambientes  
de Desenvolvimento de Processos de  
Software Utilizando a Análise da  
Sensibilidade**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de Mestre em Ciência  
da Computação

Prof. Dr. Daltro José Nunes  
Orientador

Porto Alegre, dezembro de 2011.

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Dertzbacher, Juliano

Um Modelo de Simulador para Ambientes de Desenvolvimento de Processos de Software Utilizando a Análise da Sensibilidade / Juliano Dertzbacher – Porto Alegre: Programa de Pós-Graduação em Computação, 2010.

106 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2011. Orientador: Prof. Dr. Daltro José Nunes.

1.Processos de Software 2.Tecnologia de Processos de Software 3.PSEE 4.Simulação 5.Simulador 6.Análise da Sensibilidade. I. Nunes, Daltro José. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## Agradecimentos

Inicialmente, agradeço a Deus por ter me concedido forças para persistir e concluir este trabalho.

À minha namorada Aline que compreendeu a importância deste período em nossas vidas, mantendo-se atenciosa e compreensiva com minhas constantes ausências e mau humor, me contagiando com seu amor, alegria e energia positiva.

Aos meus pais Nadir e Pedro, responsáveis pela minha formação moral, que sempre confiaram no meu potencial e sonharam com a realização desta etapa.

Ao meu irmão Guilherme pelo apoio e incentivo durante este árduo período.

A todos meus amigos que me deram apoio e acreditaram na minha capacidade para a realização deste trabalho.

Ao meu orientador Prof. Daltro e ao mestre Prof. Lincoln pelas oportunidades, formação acadêmica e ensinamentos repassados.

Aos meus colegas de trabalho do Laboratório de Engenharia de Software pelas contribuições em discussões e companheirismo nos anos em que trabalhei neste.

Aos professores do Programa de Pós-Graduação em Ciência da Computação pelo conhecimento adquirido nas disciplinas ministradas no decorrer do curso.

Finalmente, a todos aqueles que de uma ou outra forma contribuíram para a realização deste trabalho. A todos vocês o meu mais sincero obrigado!

# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>6</b>
<b>LISTA DE FIGURAS.....</b>	<b>7</b>
<b>LISTA DE TABELAS .....</b>	<b>8</b>
<b>RESUMO.....</b>	<b>9</b>
<b>ABSTRACT .....</b>	<b>10</b>
<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>1.1 Problema .....</b>	<b>13</b>
<b>1.2 Objetivos.....</b>	<b>13</b>
<b>1.3 Organização do texto.....</b>	<b>14</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
<b>2.1 Processos de software .....</b>	<b>15</b>
<b>2.2 Simulação .....</b>	<b>16</b>
2.2.1 Simulação estocástica .....	19
<b>2.3 Análise da Sensibilidade.....</b>	<b>20</b>
2.3.1 Análise da sensibilidade e simulação .....	21
<b>3 REVISÃO SISTEMÁTICA E ESTUDO SOBRE SIMULADORES DE PROCESSOS DE SOFTWARE .....</b>	<b>23</b>
<b>3.1 Revisão sistemática .....</b>	<b>23</b>
<b>3.2 Avaliação comparativa dos simuladores .....</b>	<b>25</b>
3.2.1 Simuladores avaliados .....	25
3.2.2 Cenário utilizado para a avaliação comparativa .....	28
3.2.3 Resultados da avaliação comparativa .....	28
3.2.4 Análise da avaliação comparativa .....	33
<b>4 MODELO PROPOSTO E ABORDAGEM UTILIZADA PARA O DESENVOLVIMENTO DO SIMULADOR.....</b>	<b>35</b>
<b>4.1 WebAPSEE .....</b>	<b>35</b>
<b>4.2 Modelo proposto .....</b>	<b>39</b>
<b>4.3 Implementação do APSEE- Sim.....</b>	<b>41</b>
<b>5 ESTUDO DE CASO, RESULTADOS E AVALIAÇÃO DO SIMULADOR..</b>	<b>47</b>
<b>5.1 Detalhamento do Estudo de caso.....</b>	<b>47</b>
<b>5.2 Resultados obtidos com as simulações determinísticas e estocásticas .....</b>	<b>53</b>
<b>5.3 Análise da sensibilidade a partir dos resultados obtidos .....</b>	<b>63</b>
<b>5.4 Avaliação dos recursos oferecidos pelo simulador .....</b>	<b>67</b>
<b>6 CONCLUSÃO.....</b>	<b>72</b>
<b>6.1 Contribuições .....</b>	<b>72</b>
<b>6.2 Limitações .....</b>	<b>73</b>
<b>6.3 Trabalhos futuros .....</b>	<b>73</b>
<b>REFERÊNCIAS .....</b>	<b>75</b>

<b>ANEXO A – RESULTADOS OBTIDOS COM A REVISÃO SISTEMÁTICA DOS TRABALHOS PUBLICADOS SOBRE SIMULAÇÃO DE PROCESSOS DE SOFTWARE .....</b>	<b>79</b>
<b>ANEXO B - DESCRITIVO DAS PRINCIPAIS CLASSES DO SIMULADOR ....</b>	<b>85</b>

## **LISTA DE ABREVIATURAS E SIGLAS**

PSEE	Process-centred Software Engineering Environment
RUP	Rational Unified Process
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
MPSBR	Melhoria de Processos do Software Brasileiro
ISO	International Organization for Standardization
IEEE	Institute of Electrical and Electronics Engineers
ACM	Association for Computing Machinery
SPS	Simulação de Processos de Software
UML	Unified Modeling Language
UFPA	Universidade Federal do Pará
UFRGS	Universidade Federal do Rio Grande do Sul
QAG	Quantidade de agentes
SCH	Soma do custo por hora
MHA	Média do grau de habilidade

## LISTA DE FIGURAS

Figura 2.1: Estudo de caso simplificado utilizado no processo de avaliação.....	28
Figura 4.1: Ciclo de vida de processos de software (LIMA, 2003).....	36
Figura 4.2: Visão geral do modelo WebAPSEE (LIMA, 2003).....	37
Figura 4.3. Modelo proposto para o simulador. ....	40
Figura 4.4: Casos de uso do simulador.....	41
Figura 4.5: Diagrama entidade relacionamento do simulador.....	43
Figura 4.6: Diagrama de atividades do simulador.....	44
Figura 4.7: Diagrama de classes do simulador.....	45
Figura 4.8: Interface do APSEE-Sim .....	46
Figura 5.1: Estudo de caso modelado no WebAPSEE.....	48
Figura 5.2: Resultados obtidos na simulação do tempo em dias.....	62
Figura 5.3: Resultados obtidos na simulação do custo em reais. ....	62
Figura 5.4: Análise da sensibilidade do tempo em dias .....	66
Figura 5.5: Análise da sensibilidade do custo em reais.....	66
Figura 5.6: Equações elaboradas no <i>Easy Eclipse</i> .....	68
Figura 5.7: <i>Mozilla Firefox</i> com os recursos do APSEE-Sim.....	68

## LISTA DE TABELAS

Tabela 2.1. Comparativo entre as ferramentas avaliadas, relativo à modelagem.....	30
Tabela 2.2. Comparativo entre as ferramentas avaliadas, relativo à simulação. ....	32
Tabela 2.3. Comparativo entre as ferramentas avaliadas, relativo aos resultados.....	33
Tabela 5.1: Resultados da simulação determinística. ....	54
Tabela 5.2: Resultados da simulação estocástica. ....	55
Tabela 5.3: Resultado da análise da sensibilidade em função da quantidade de agentes (QAG).....	64
Tabela 5.4: Resultado da análise da sensibilidade em função da soma do custo por hora (SCH).....	65
Tabela 5.5: Resultado da análise da sensibilidade em função da média do grau de habilidades (MHA).....	65
Tabela 5.6: Características avaliadas no APSEE-Sim.....	70



## RESUMO

A construção de um software envolve alto grau de risco e exige do gerente muito planejamento para atender as estimativas orçamentárias e cumprir os prazos estipulados. No contexto dos processos de software, são escassos os recursos tecnológicos que permitam extrair conhecimento dos processos modelados nos PSEEs e apontem quais fatores provocam os maiores impactos no resultado final, fornecendo novas perspectivas para melhorar a gerência. Para suprir estas carências, é possível utilizar a simulação na obtenção de informações sobre as atividades do processo e a análise da sensibilidade na identificação das variáveis que influenciam de forma mais significativa nos resultados. Neste sentido, este trabalho propõe um modelo de simulador, integrado a uma ferramenta de apoio à gerência de projetos centrados em processos, que utiliza os dados da base do PSEE, oferece recursos para manipular os dados do processo de forma determinística ou estocástica (simulação), permite testar vários cenários e possibilita a análise de quais variáveis impactam de forma mais significativa no resultado final (análise da sensibilidade), antes de iniciar a execução das atividades. O desenvolvimento do modelo de simulador foi fundamentado nos conhecimentos adquiridos com a revisão sistemática dos trabalhos publicados sobre simulação nos últimos anos e também na avaliação comparativa dos recursos tecnológicos oferecidos pelas ferramentas identificadas nas publicações selecionadas na revisão. Os resultados obtidos com a implementação do modelo proposto, utilizando as informações de um estudo de caso real, modelado no WebAPSEE, forneceram informações que indicam melhorias em relação ao custo e ao tempo de desenvolvimento do processo em estudo, assim como a identificação da variável de maior sensibilidade, permitindo otimizar a execução destas atividades.

**Palavras-Chave:** Processos de Software, Tecnologia de Processos de Software, PSEE, Simulação, Simulador, Análise da Sensibilidade.

## **ABSTRACT**

The construction of a software involves high degree of risk and requires of the manager a lot of planning to attend the budget estimates and meet deadlines. In the context of software processes, there are limited technological resources that allow the extract of knowledge of the processes modeled in PSEEs and indicate what factors cause the greatest impact on the final result, providing new opportunities to improve management. To overcome these deficiencies, is possible to use simulation to obtain information about the activities of the process and sensitivity analysis to identify the variables that most significantly influence the results. Thus, this work proposes a simulation model, integrated to a tool that support project management centered processes, which uses data from the base of the PSEE, offers resources to handle the process data in a deterministic or stochastic way (simulation), allows the testing of various scenarios and enables the analysis of which variables most significantly impact the final result (sensitivity analysis), before starting the execution of activities. The simulation model development was based on the knowledge gained with a systematic review of the papers on simulation in recent years and a comparative evaluation of technological resources offered by the tools identified in the selected papers in the review. The results obtained with the implementation of the proposed model, using the information in a real case study, modeled on WebAPSEE, provided information that indicates improvements in relation to the cost and time development of the process under study, as well as the identification of the variable with the greatest sensitivity, allowing to optimize the performance of these activities.

**Keywords:** Software Processes, Software Process Technology, PSEE, Simulation, Simulator, Sensitivity Analysis.

# 1 INTRODUÇÃO

As evoluções tecnológicas vividas por nossa sociedade nos últimos anos têm evidenciado o valor da informação e provocado uma utilização crescente de software capaz de processar a enorme quantidade de dados utilizados, destacando-se como instrumento estratégico para a diferenciação de produtos e serviços. A presença de sistemas de software em diversos domínios de negócio, aplicando diferentes tecnologias e com requisitos complexos, resultou no aumento contínuo do seu tamanho e complexidade de desenvolvimento (FUGGETTA, 2000).

Apesar dos inúmeros avanços recentes relacionados ao desenvolvimento de software, esta atividade possui um alto grau de risco. O desenvolvimento de software já gerou grandes prejuízos no passado e continua gerando. Atualmente, muitos projetos de desenvolvimento de software são iniciados e não são terminados, e outros são terminados consumindo prazos e orçamentos bem acima do que foi estipulado no início do projeto. Além disso, muitos produtos desenvolvidos possuem um nível muito baixo de qualidade. Assim, surgiram diversos estudos sobre a concepção de técnicas e métodos para entender e melhorar a qualidade do desenvolvimento do software. Neste contexto, os estudos sobre Processos de Software têm se destacado (FUGGETTA, 2000).

Um processo de software fornece uma visão inclusiva e integrativa, pois relaciona diversos elementos gerenciáveis (i.e., atividades, dependências, agentes, recursos, etc.) que indicam quais os passos a serem executados, quem os executará, como, quando, onde, quais os recursos serão aplicados, e o porquê de sua realização. Esta visão ampliada do desenvolvimento fornece uma infraestrutura adequada a teorias e práticas capazes de acomodar novas demandas, tais como o desenvolvimento gerenciável com foco sistêmico e organizacional e a avaliação do impacto das mudanças no processo.

Os avanços nos estudos na área resultaram no desenvolvimento de modelos de processo, como por exemplo, Espiral, RUP e V-Model; modelos de capacitação de processos, como CMM, CMMI, Spice e MPSBR; normas como ISO9001/9000-3, ISO/IEC15504 e; os ambientes de desenvolvimento de software orientados a processos como Spearmint (RÖSCH, ZETTEL, *et al.*, 1998), Dynamite (HEIMANN, JOERIS, *et al.*, 1996), ProcessWeaver (PROCESSWEAVER, 2011) e WebAPSEE (WEBAPSEE, 2011). Estes resultados têm revolucionado a maneira como as organizações planejam, executam e gerenciam o desenvolvimento de software.

Apesar dos inúmeros avanços relacionados ao desenvolvimento de software, são frequentes os relatos sobre problemas, prejuízos financeiros e acidentes causados por falhas no sistema. Assim, surgiram diversos estudos sobre a concepção de técnicas e métodos para entender e melhorar a qualidade dos processos de desenvolvimento de software (GRUHN, 2002). A literatura especializada na área (FUGGETTA, 2000),

(BEECHAM, HALL e RAINER, 2003), (CURTIS, 2000), (MÜNCH, ROMBACH e RUS, 2003), (ZHANG, KITCHENHAM e PFAHL, 2008), evidencia a necessidade de estudos em vistas de um melhor entendimento sobre os fenômenos relacionados ao processo, às pessoas e ao negócio, que afetam o desenvolvimento de software.

Embora o software tenha importância primordial para o êxito do mercado de alta tecnologia e em todas as áreas de serviço, as práticas da engenharia de software ainda não correspondem a este desafio (PFAHL, RUHE e KOVAL, 2001).

Como reflexo do escasso conhecimento sobre processos, a seguir são apresentadas algumas perguntas que ainda não foram totalmente esclarecidas: Qual a razão do sucesso/fracasso de um determinado projeto? Como acelerar o projeto sem que a qualidade seja influenciada negativamente? Como adaptar adequadamente um processo a um contexto específico, preservando a produtividade e a qualidade? Quais métodos, técnicas, e ferramentas produzem melhor resultado em um determinado contexto de desenvolvimento?

Desta forma, diversos autores (BASILI, 1996), (KITCHENHAM, PFLEEGER, *et al.*, 2002), (PERRY, PORTER e VOTTA, 2000) discutem a importância dos Estudos Empíricos (i.e., experimentação, estudos de casos, etc.) para o processo de aquisição de conhecimento sobre o processo de software e os fenômenos relacionados. Contudo, a realização de estudos empíricos, especialmente a experimentação, exige muito esforço, tempo e dinheiro. Recentemente, tem sido amplamente incentivada a combinação da simulação com os estudos empíricos.

Através da simulação é possível minimizar os obstáculos originalmente presentes nos estudos empíricos tradicionais. É possível representar a estrutura e o comportamento do problema a ser investigado, reduzindo a necessidade da realização de experimentos reais.

Segundo (KELLNER, MADACHY e RAFFO, 1999) a simulação pode ser utilizada sempre que a complexidade do sistema a ser modelado for além da possibilidade de representação estática do modelo. Por exemplo: em domínios onde existe pouca experiência sobre o processo, a relação de causa-efeito sobre o produto não é muito clara. Assim, obter um entendimento detalhado sobre essa relação requer muitas execuções do processo sobre diferentes condições, e o registro dos resultados para determinar um melhor entendimento sobre os parâmetros que influenciam a saída do processo. Na prática isso não é possível, pois demanda muito tempo e dinheiro e envolve inúmeros desenvolvedores.

Com a simulação de processos de software, é possível reduzir custos, tempo e aprimorar o conhecimento do processo. A redução do custo pode ser obtida com a substituição dos experimentos reais pela simulação. A redução do tempo acontece pelo fato da simulação poder ser realizada quantas vezes for necessária, com velocidade desejada e relativamente inferior a da experimentação real. Em relação à melhoria de conhecimento, a simulação pode ser utilizada para propósitos de treinamento, e seus resultados podem, em alguns casos, ser replicados em diferentes contextos.

De acordo com (RAFFO, 2000), no contexto do processo de desenvolvimento de software, a simulação tem sido utilizada com o propósito de avaliar tecnologias, auxiliar na tomada de decisões sobre a gerência de projetos, planejamento de projetos, entendimento e aprendizado de processos de software, melhoria de processos, gerenciamento estratégico do processo de software e treinamento.

## 1.1 Problema

A fim de reduzir custos e cumprir os prazos de desenvolvimento de projetos de software, é essencial que os gerentes planejem e agendem da melhor forma os seus projetos. Estimar tempo, custo, qualidade, riscos e outras, não é uma tarefa fácil. É comum ocorrer atrasos e valores incorretamente orçados ou mal aferidos nos projetos de software devido a imprevistos e retrabalho provocado pelo *feedback* obtido no desenvolvimento (PARK, CHOI, *et al.*, 2007).

Problemas de planejamento, gerência de projeto, qualidade de software, cumprimento dos prazos e custos, entre outros, podem ser contornados com o uso da simulação. E, neste contexto, o apoio tecnológico disponível para o desenvolvimento da simulação é fator determinante para o sucesso do trabalho.

Levando em consideração o fato de que uma das principais razões para o fracasso de projetos de desenvolvimento de software está relacionada a deficiências em sua gerência (PARK, CHOI, *et al.*, 2007), o problema a ser resolvido neste trabalho refere-se à escassez de recursos tecnológicos para a simulação dos processos de software de um PSEE<sup>1</sup>, possibilitando extrair conhecimento destes e apontar quais variáveis provocam os maiores impactos no resultado final, fornecendo novas perspectivas para melhorar a sua gerência.

## 1.2 Objetivos

Para que este problema possa ser resolvido, este trabalho tem como o objetivo geral propor um modelo de simulador incorporado a um PSEE (conceito), que ao ser implementado (prova de conceito) ofereça ao gerente os seguintes recursos (objetivos específicos):

- Capturar informações do processo (variáveis);
- Elaborar equações sobre um determinado domínio (tempo, custo ou outro);
- Manipular os dados de forma determinística (levando em consideração somente os valores já modelados ou valores fixos) ou estocástica (gerando valores aleatórios com a técnica Monte Carlo);
- Exportar e analisar as informações obtidas;
- Identificar as variáveis que têm maior impacto sobre o resultado final (análise da sensibilidade) e sobre os diferentes cenários, para que este escolha aquele que forneça os melhores resultados ou que melhor se adapte à realidade do projeto.

Com a aplicação da simulação no contexto da tecnologia de processo de software (PSEEs), e utilizando a análise da sensibilidade para avaliar o modelo de simulador, esta solução viabiliza ao gerente de projetos de desenvolvimento de software avaliar laboratorialmente diferentes cenários de desenvolvimento durante a fase de planejamento do projeto. Desta forma, a solução proposta tende a contribuir para o avanço das pesquisas na área da engenharia de software, especialmente na simulação de processos de software.

---

<sup>1</sup> do inglês: Process-Centered Software Engineering Environment

### **1.3 Organização do texto**

O restante do texto está organizado como segue:

O capítulo 2 apresenta a fundamentação teórica acerca dos conceitos abordados na área que está sendo explorada. O capítulo 3 apresenta um estudo sobre os simuladores de processos de software, bem como uma avaliação de acordo com critérios pré-estabelecidos. O capítulo 4 apresenta detalhadamente o modelo proposto. O capítulo 5 apresenta uma avaliação da proposta. E, finalmente, o capítulo 6 apresenta as conclusões do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

A metodologia empregada neste trabalho abrange o estudo e aplicação prática de diferentes conceitos da Engenharia de Software (processos de software, simulação e análise da sensibilidade) com o propósito de apoiar a construção de uma solução de software que viabilize a experimentação de processos de software pela simulação de processos, utilizando a análise da sensibilidade. As subseções a seguir descrevem o relacionamento deste projeto com as áreas envolvidas.

### 2.1 Processos de software

Um processo de software pode ser definido como um conjunto de atividades ordenadas que tem, como objetivo, alcançar um produto de software. Um modelo de processos de software é uma descrição formal do processo de software e tem, como função, indicar quais os passos a serem executados, quem os executará, como, quando, onde e o porquê de sua realização (LIMA, 2003). Um modelo é constituído por atividades (por exemplo: análise, projeto, revisão, etc.) que precisam ser realizadas por agentes ou grupos (humanos ou não) que produzem ou consomem artefatos (diagramas de análise, algoritmos, manuais, etc.), e/ou recursos (software, equipamentos técnicos, etc.), necessários para a realização da atividade.

Estudos recentes sobre a qualidade de software, na sua maioria, são voltados para o melhoramento do processo de desenvolvimento. Isto não significa que a qualidade do produto não seja importante, o fato é que, ao garantir a qualidade do processo, já se está dando um grande passo para se garantir também a qualidade do produto (LONCHAMP, 1993).

A área de tecnologia de processo de software representou um importante passo em direção à melhoria da qualidade do software. Diversas teorias, conceitos, formalismos, metodologias e ferramentas surgiram, enfatizando a descrição formal do modelo de processo de software (LIMA, 2003). Com o uso de ambientes para a automação do desenvolvimento de processo de software (PSEEs), pode-se coordenar atividades de equipes geograficamente dispersas, acompanhar os prazos e consumo de recursos, além de facilitar a reutilização de boas práticas gerenciais por diferentes projetos adotados por uma organização.

Neste sentido, diversas técnicas são utilizadas para analisar o desenvolvimento do processo de software. Contudo, nos últimos anos, a simulação do processo de software tem recebido uma atenção especial por permitir uma melhor compreensão do processo e oferecer suporte aos problemas relacionados às decisões de gerenciamento (KIRK, 2004).

A utilização de modelos de simulação de processos de software surge como uma proposta interessante. Através desses modelos, é possível representar o comportamento

de gerentes e desenvolvedores em um ambiente de desenvolvimento de software baseado em um modelo de processos. Desta forma, é possível prever de forma aproximada as consequências que a aplicação de um modelo terá para um ambiente de desenvolvimento. A possibilidade de representar os passos de desenvolvimento, utilizando modelos de processo de software e, a possibilidade de testar esses modelos antes que eles sejam efetivamente implantados, através da simulação de projetos de software, permite quantificar o que está sendo representado e descobrir possíveis inconsistências que poderiam acarretar grandes prejuízos se ocorressem em um projeto real (DORES, 2001).

## 2.2 Simulação

O principal fundamento para utilizar a simulação em qualquer assunto é o incessante desejo do homem de conhecer o futuro. O verbo “simular” surgiu para descrever a arte de construção de modelos e, a partir de 1950, a simulação assumiu outro significado, tornando possível a realização de experiências com modelos matemáticos em computadores. Pela primeira vez os cientistas sociais chegaram à conclusão de que poderiam conduzir experiências de laboratório controladas (NAYLOR, BALINTFY, *et al.*, 1966).

A simulação é essencialmente uma técnica que consiste em criar um modelo da situação real e testar nele diferentes experiências. Este modelo normalmente toma a forma de um conjunto de considerações relacionadas à operação do sistema. Estas considerações são expressas através de relações matemáticas, lógicas e simbólicas entre as entidades, ou objetos de interesse, do sistema. Uma vez construído e validado, um modelo pode ser usado para investigar uma grande quantidade de questões do tipo “e se” sobre o sistema do mundo real. Alterações no sistema podem ser inicialmente simuladas para se prever as consequências no mundo real. A simulação também pode ser usada para estudar sistemas no estágio de projeto, ou seja, antes do sistema ser construído.

Segundo (NAYLOR, BALINTFY, *et al.*, 1966), pela simulação em computadores, pode-se penetrar na complexidade dos sistemas, construir e testar hipóteses. A escolha final de uma ação ou teoria que se aplique à realidade ainda permanece com os especialistas, mas ela é capaz de fornecer poderosa assistência como instrumento analítico.

Na Engenharia de Software é fácil propor hipóteses, contudo, é muito difícil testá-las. Experimentos controlados são onerosos e quase impossíveis de se realizar em condições reais. Além disso, o efeito isolado e a avaliação do impacto de um determinado fator dentro de contexto maior, complexo e dinâmico, pode ser extraordinariamente difícil (WERNICK e HALL, 2007). Desta forma, o apoio tecnológico disponível para o desenvolvimento da simulação é fator determinante para o sucesso do estudo.

Neste contexto, a simulação pode ser realizada satisfatoriamente sempre que for impossível ou extremamente oneroso observar certos processos no mundo real. É possível usar a simulação como um meio efetivo de gerar dados numéricos descritivos de processos que, de outra maneira, só poderiam fornecer tais informações a um preço muito elevado. O sistema observado também pode ser tão complexo que se torne impossível descrevê-lo em termos de um conjunto de equações matemáticas e para o qual seja possível obter soluções analíticas, que enfim seriam usadas para fins



preditivos. Mesmo que um modelo matemático possa ser desenvolvido para descrever um sistema de interesse, pode não ser possível obter uma solução para o modelo, através de técnicas analíticas diretas e, em consequência, prever o comportamento do sistema (WERNICK e HALL, 2007).

Ainda que a principal razão para a escolha da simulação em computadores seja sua capacidade de superar as dificuldades existentes na execução do modelo, segundo (KELLNER, MADACHY e RAFFO, 1999), existem outras razões para se usar a simulação. A simulação permite estudar e experimentar complexas interações internas de um sistema.

- Através da simulação pode-se estudar os efeitos de certas variações da organização ou das informações relativas à operação de um sistema, fazendo-se alterações no seu modelo e observando os efeitos dessas alterações no comportamento do sistema;
- As observações detalhadas do sistema que está sendo simulado podem conduzir à sua melhor compreensão e às sugestões para melhorá-lo;
- A simulação pode ser utilizada como material pedagógico, para ensinar habilidades básicas na análise teórica, na análise estatística e na arte de decidir, tanto a estudantes, como a profissionais;
- A experiência em projetar modelos de simulação pode ser mais útil que a própria simulação. O conhecimento obtido em projetar um estudo de simulação frequentemente sugere variações no sistema que está sendo simulado. Os efeitos dessas variações podem então ser testados via simulação antes de serem executados no sistema real;
- Simular sistemas complexos pode fornecer valiosas informações no sentido de quais são as variáveis mais importantes do sistema e como essas variáveis interagem;
- A simulação pode ser usada para experiências com novas situações sobre as quais se tem pouco ou mesmo nenhuma informação;
- A realização de uma simulação pode servir como um primeiro teste para se delinear novas políticas e regras de decisão para a operação de um sistema, antes de se correr o perigo de experimentá-las no sistema real;
- As simulações podem ser valiosas por propiciar uma maneira conveniente de se desdobrar um sistema complicado em subsistemas, cada um dos quais pode ser elaborado por um analista ou por uma equipe especialista naquela área;
- Para certos tipos de problemas estocásticos a sequência de eventos pode ser de particular importância. Informações sobre valores ou momento esperados podem não ser suficientes para descrever o processo. Nesses casos o método Monte Carlo pode ser a única maneira satisfatória de prover as informações necessárias;
- A simulação Monte Carlo pode ser realizada para verificar soluções analíticas;
- A simulação permite o estudo de sistemas dinâmicos quer em tempo real, em tempo reduzido ou dilatado;
- Quando novos componentes são introduzidos num sistema, a simulação pode ser usada para ajudar a prever áreas críticas e outros problemas que podem surgir na operação do sistema;

- A simulação generaliza mais que especializa. Os analistas são forçados a uma apreciação a uma apreciação e entendimento de todas as facetas do sistema, resultando que as conclusões são menos propensas a serem influenciadas por inclinações particulares e menos propensas à impraticabilidade dentro do sistema de referência.

Contudo, algumas desvantagens também devem ser consideradas:

- A construção de modelos de simulação requer treinamento especial. É uma arte que é aprendida com tempo e experiência. Além disto, se dois modelos são construídos por dois profissionais competentes, eles terão semelhanças, mas será altamente improvável que sejam iguais;
- Os resultados de uma simulação podem ser difíceis de interpretar. Como a maioria das saídas de uma simulação são variáveis aleatórias (elas estão normalmente baseadas em entradas aleatórias), é difícil determinar se uma observação é o resultado do relacionamento entre as variáveis do sistema ou consequência da própria aleatoriedade;
- A construção e análise de modelos de simulação pode consumir muito tempo e, como consequência, muito dinheiro. Economizar por sua vez pode levar a modelos incompletos;
- A simulação é usada em muitos casos onde uma solução analítica é possível;
- A simulação não dá resultados exatos.

Diferentes sistemas taxionômicos foram sugeridos para classificar os modelos de simulação, de acordo com (NAYLOR, BALINTFY, *et al.*, 1966). Este sistema classifica os modelos de simulação em determinísticos, estocásticos, estáticos e dinâmicos.

Nos modelos determinísticos, não se permite às variáveis exógenas (descrevem atividades e eventos que não ocorrem dentro do sistema) e endógenas (descrevem atividades e eventos que ocorrem dentro do sistema) serem variáveis randômicas, e as características operacionais devem ser relações exatas e não funções de densidade de probabilidade. Os modelos determinísticos são, computacionalmente, menos exigentes que os estocásticos e podem frequentemente ser resolvidos por técnicas como a do cálculo de máximos e mínimos (NAYLOR, BALINTFY, *et al.*, 1966).

Os modelos estocásticos são assim chamados quando pelo menos uma das características operacionais é dada por uma função de probabilidade. O uso desta técnica analítica é muito limitada para a solução destes modelos porque eles são consideravelmente mais complexos que os modelos determinísticos. Por isso, a simulação é mais adequada como método de análise e solução para modelos estocásticos do que para os determinísticos. Os modelos estocásticos são também de interesse do ponto de vista da geração de amostras randômicas de dados para serem usados nos estágios de observação ou de testes de averiguação científica (NAYLOR, BALINTFY, *et al.*, 1966).

Modelo estático é aquele que leva em conta explicitamente a variável tempo. A maioria dos modelos estáticos são completamente determinísticos e as soluções podem ser normalmente obtidas por meio de técnicas analíticas diretas, tais como o cálculo de otimização e a programação matemática (NAYLOR, BALINTFY, *et al.*, 1966).

Os modelos de simulação dinâmicos podem ser discretos ou contínuos. Em uma simulação discreta, que também pode ser chamada de evento-discreta, são considerados somente os eventos onde há alteração do sistema, ou seja, o tempo decorrido entre

alterações do estado do sistema não é relevante para a obtenção dos resultados da simulação, embora o tempo nunca pare. Na simulação contínua, o sistema se altera a cada fração de tempo (NAYLOR, BALINTFY, *et al.*, 1966).

### 2.2.1 Simulação estocástica

A análise Monte Carlo é uma técnica de simulação para problemas que tem base probabilística ou estocástica. Segundo (NAYLOR, BALINTFY, *et al.*, 1966), os problemas que envolvem alguma forma de processo estocástico dão margem ao uso desta técnica. A demanda de consumidores, o tempo de produção e o investimento total de uma organização são exemplos de variáveis econômicas que podem ser consideradas de natureza estocástica. Os métodos Monte Carlo foram desenvolvidos não só para simular a maioria das distribuições de probabilidades como para o caso das distribuições empíricas; certos problemas matemáticos completamente determinísticos não podem ser facilmente resolvidos (se a solução houver) por métodos estritamente determinísticos. Entretanto, é possível obter soluções aproximadas para esses problemas simulando um processo estocástico cujos momentos, funções de densidade ou funções de distribuição cumulativas satisfaçam as relações funcionais ou requisitos de solução de problemas determinísticos. Soluções de equações diferenciais e problemas de integrais múltiplas podem ser obtidas, às vezes, mais rapidamente pelo uso deste método de análise numérica do que por outros métodos.

Embora existam métodos analíticos para a resolução de problemas matemáticos determinísticos, os métodos de simulação estocástica têm-se mostrado mais eficientes na obtenção de soluções para esses problemas do que outros métodos mais ortodoxos. Contudo, é importante ressaltar que a simulação constitui mera técnica de análise numérica e deve ser preferida somente se sua eficiência relativa na obtenção de soluções numéricas for superior a de outras técnicas.

As soluções através dos métodos de simulação estocástica aparece em situações onde seria indicado um método de amostragem estatística, porém onde a amostragem seria impossível ou economicamente desaconselhável. Uma simulação estocástica acarreta a construção de um modelo probabilístico do processo a ser estudado, enquanto que a experiência de amostragem simples é usualmente executada sobre os dados brutos (NAYLOR, BALINTFY, *et al.*, 1966).

A geração de estatísticas simuladas (valores randômicos) é de natureza inteiramente numérica e é feita introduzindo-se números pseudo-randômicos no processo ou sistema em estudo (onde o sistema é representado por um modelo probabilístico) e obtendo-se números (valores randômicos) do mesmo como respostas. Como regra geral, a simulação estocástica envolve a substituição de um universo estatístico (de elementos) real, pelo seu correlativo teórico, um universo descrito por uma determinada distribuição de probabilidades (por exemplo, a distribuição normal) e, a seguir, uma amostragem dessa população teórica por meio de um tipo qualquer de gerador de números randômicos (NAYLOR, BALINTFY, *et al.*, 1966).

Contudo, em alguns casos pode não ser possível encontrar uma distribuição teórica padrão que descreva um processo estocástico particular, ou algum componente de tal processo. Nesses casos, o processo estocástico pode ser reproduzido (ou simulado) apenas através de amostragem de distribuições empíricas e não por intermédio de alguma distribuição teórica bem conhecida (NAYLOR, BALINTFY, *et al.*, 1966).

É aconselhável considerar primeiramente a utilização de distribuições teóricas padrão. Se nenhuma das distribuições padrão descrever adequadamente o comportamento do processo, então devemos, necessariamente, recorrer a distribuições empíricas.

### 2.3 Análise da Sensibilidade

A análise da sensibilidade é um método matemático para calcular as mudanças no comportamento do sistema devido às mudanças nos parâmetros ou variáveis. As incertezas na estimativa e utilização de parâmetros são comuns na modelagem dos sistemas. Em muitos casos, quantificar exatamente os valores, representados pelos parâmetros do modelo, é difícil ou impossível. Esse método oferece um meio de corrigir esse problema, quantificando matematicamente o comportamento do modelo para pequenas mudanças em seus parâmetros. Uma pequena mudança em um parâmetro, resultando em uma grande mudança no resultado do modelo mostra uma alta influência (sensibilidade) do parâmetro sobre o comportamento do modelo. Neste caso, a estimativa do parâmetro precisa ser a mais precisa possível. A análise da sensibilidade permite determinar a precisão dos parâmetros que compõem o modelo. Além de detectar os parâmetros críticos do modelo, ela também fornece uma estimativa otimista e pessimista dos resultados do projeto com base nas alterações dos parâmetros de entrada (CANGUSSU, DECARLO e MATHUR, 2003).

Formalmente, uma solução é considerada de alta sensibilidade se a ocorrência de uma mudança nos coeficientes originais provoca alterações na equação. Neste contexto, avaliar o impacto das entradas em relação às saídas é muito importante. Para que uma análise da sensibilidade possa ser desenvolvida satisfatoriamente é necessário obter todos os dados referentes a um problema ou coletar uma grande quantidade de informações, resultantes da experimentação de diversos cenários, que possam dar uma ideia das saídas a serem geradas (CHINNECK, 2006).

A análise da sensibilidade na Ciência da Computação visa melhorar a estabilidade das aplicações, considerando a incerteza gerada pelas perturbações nos parâmetros. Este método pode ser utilizado para investigar a estabilidade e as propriedades de sensibilidade dos programas, tais como aqueles utilizados em sistemas de controle em tempo real. Quando um procedimento computacional é instável, mudanças muito pequenas nas entradas poderão causar grandes mudanças nas saídas. Procedimentos computacionais instáveis levam a resultados questionáveis. Além disso, quando um procedimento computacional tem muitos parâmetros, a análise da sensibilidade também pode ser utilizada para determinar quais parâmetros causam as maiores alterações nos resultados (LIEBROCK, 2005).

Segundo (LIEBROCK, 2005), a estabilidade refere-se ao comportamento do programa. Em relação ao comportamento, se houver uma pequena alteração na entrada, então:

- Há uma mudança relativamente pequena na saída, o que é aceitável. Neste caso pode-se dizer que o procedimento computacional é estável, ou;
- Há uma mudança drástica na produção, que por sua vez pode levar a dois resultados: Essa alteração é aceitável (isto poderia representar corretamente o modelo) ou essa alteração é inaceitável, esta situação não representa corretamente o modelo e indica um erro. Nestes casos, é possível afirmar que o código é instável. No entanto, alguns desenvolvedores salientam que a

instabilidade também ocorre na natureza e, para modelar a natureza com fidelidade, a instabilidade também deve ser modelada. As mudanças drásticas na saída, resultantes de pequenas mudanças nas entradas, devem ser estudadas para garantir que o modelo reflita o comportamento do sistema simulado.

Os resultados obtidos com a análise da sensibilidade podem ser utilizados para diversos fins, como, por exemplo, a classificação das entradas na ordem de seu impacto sobre a produção, avaliação de mudanças na produção devido às variações nos parâmetros de entrada, melhoria da qualidade dos cálculos ou limitação na utilização de um programa para as partes relevantes no domínio de entrada.

Existem várias abordagens da análise da sensibilidade. Caso o modelo seja simples o suficiente para ser resolvido rapidamente, pode ser usada a abordagem força bruta, alterando-se os dados iniciais e resolvendo-se o modelo diversas vezes para ver quais resultados são obtidos. Se o modelo for complexo e exigir muito tempo para ser resolvido, podem-se aplicar os métodos formais da análise da sensibilidade clássica. Os métodos clássicos dependem da relação entre os parâmetros iniciais e finais (em especial os parâmetros ótimos) para atualizar rapidamente a melhor solução quando as mudanças são feitas para os coeficientes originais. Entre esses dois extremos está a abordagem *computer-based ranging*. A informação sobre o quanto determinados coeficientes podem mudar, antes da solução ótima, é fundamentalmente alterada. A maioria das soluções comerciais fornece estas informações (CHINNECK, 2006).

### **2.3.1 Análise da sensibilidade e simulação**

Uma simulação válida cria um eficaz e preciso modelo de sistema físico real, que descreve suas granularidades em interações, dinâmicas e incertezas, suficientes para permitir a experimentação com a entrada de diversos cenários e hipóteses sobre o modelo de simulação. Baseado na simulação válida, a análise da sensibilidade visa obter *insights* e novos conhecimentos sobre o sistema real, que acabam por melhorar a relação custo-benefício e eficiência do sistema. Muitas vezes, os procedimentos que definem a saída ideal são auxiliados pelo conhecimento da sensibilidade das entradas de um modelo de simulação (LU, CHAN e YEUNG, 2005).

Para facilitar a modelagem experimental, a sensibilidade de entrada de um modelo de simulação pode ser obtida com uma análise sistemática fatorial. Essencialmente, neste procedimento, um fator é alterado de cada vez e os outros são mantidos inalterados, avaliando-se e observando-se a resposta do modelo através das alterações nas saídas, executando-se o modelo de simulação repetidamente (LU, CHAN e YEUNG, 2005).

A aplicação da análise da sensibilidade na simulação determinística é essencial, pois fornece recursos para a avaliação sistemática dos modelos. Nesta área está incluído o comportamento e a validação do modelo, estimativas quanto às incertezas do modelo, investigação do processo de decisão em face aos modelos com resultados incertos, e identificação das áreas de investigação promissoras ou não (LEHR, CALHOUN, *et al.*, 1994).

A análise da sensibilidade consiste na apreciação das mudanças geradas na saída de um sistema dinâmico, em resposta às alterações especificadas nos parâmetros de entrada e condições iniciais do sistema. Um aspecto fundamental deste método é saber diferenciar as variáveis dependentes de interesse em relação aos parâmetros do modelo

e, em seguida, realizar a expansão em série dessas variáveis sobre o valor nominal dos valores dos parâmetros. Este procedimento, que gera as chamadas funções de sensibilidade, pode ser interpretado, simultaneamente, como uma medida da sensibilidade de um modelo aos seus parâmetros e seu nível de incerteza. A equação de propagação de erro e a função de análise da sensibilidade do sistema estão intimamente relacionadas e, essencialmente, transmitem a mesma informação (LEHR, CALHOUN, *et al.*, 1994).

Uma observação sobre o estado da arte da análise da sensibilidade foi feita por (Chinneck, 2006), que apontou como limitação da técnica a capacidade de analisar o impacto de variações de apenas um coeficiente de cada vez. Existem algumas técnicas aceitas para alterar vários coeficientes de uma só vez: a regra 100% e a programação paramétrica. A regra 100% é normalmente limitada à mudança de apenas alguns coeficientes de vez, com limites do quanto eles podem variar, e a programação paramétrica muda todos os coeficientes da relação. No entanto o que é realmente necessário é um método que permita a variação de todos os coeficientes de forma independente (CHINNECK, 2006).

A fundamentação teórica apresentada neste capítulo facilita o entendimento dos estudos apresentados no próximo capítulo, acerca da revisão sistemática desenvolvida para melhor absorver os conhecimentos das publicações sobre simulação nos últimos anos e também na avaliação das ferramentas disponíveis para a simulação de processos de software.

### 3 REVISÃO SISTEMÁTICA E ESTUDO SOBRE SIMULADORES DE PROCESSOS DE SOFTWARE

Com o propósito de conhecer o passado e o presente da simulação de processos de software, uma revisão sistemática foi desenvolvida buscando identificar iniciativas e experiências na engenharia de software relacionadas ao tema. Esta revisão permitiu uma familiarização maior com a fundamentação teórica aplicada na proposta do modelo e desenvolvimento do simulador. Também foi possível identificar trabalhos relacionados que propõem ou utilizam ferramentas capazes de conduzir simulações de processos de software. Os simuladores identificados neste estudo e no trabalho de (ZHANG, KITCHENHAM e PFAHL, 2008) foram analisados e resultaram na avaliação comparativa descrita a seguir.

#### 3.1 Revisão sistemática

A revisão sistemática foi desenvolvida a fim de responder questões como: Quais as iniciativas de simulação de processos de software nos últimos anos? Qual o propósito (*for what*)? Quais motivos (*why*)? Quais as técnicas utilizadas (*how*)? Em que circunstâncias (*when*)? Quais as justificativas para o desenvolvimento de um trabalho sobre simulação de processos de software? Por que é importante publicar material sobre o assunto?

Para esclarecer estes questionamentos foram considerados artigos completos que apresentam relatos de iniciativas e experiências com simulação de processos de software e publicados nas seguintes fontes:

- IEEE Journal and Conferences (<http://www.ieeexplore.ieee.org>);
- ACM Journal and Conferences (<http://www.acm.org/dl>);
- Journal of Software Process: Improvement and Practice (<http://www3.interscience.wiley.com/cgi-bin/jhome/15482>);
- Journal of Systems and Software (<http://sciencedirect.com/science/journal/01641212>);
- Kluweronline (<http://www.springerlink.com/computer-science/>);
- Software Process Improvement (Lecture Notes in Computer Science);
- ScienceDirect (<http://www.sciencedirect.com/>).

A realização da busca pelos artigos, nas fontes mencionadas, levou em consideração as seguintes palavras chaves e sinônimos:

- Simulação de processo de software; Modelagem e simulação de processos de software.
- Simulação de projetos; Simulação de projetos de software.
- Processo de software; Processo de desenvolvimento de software.

- Engenharia de software empírica; Engenharia de software experimental.
- População: publicações relacionadas à simulação de processos de software.

Os seguintes argumentos de pesquisa também foram utilizados:

- (“software process simulation” OR “modeling and simulation of software process” OR “software process modeling and simulation” OR “process simulation” OR “system dynamics modeling”) AND (“software engineering” OR “empirical software engineering” OR “experimental software engineering” OR "software project" OR "software development")
- IEEE: (((<or> (software process simulation, modeling and simulation of software process, software process modeling and simulation, system dynamics modelling)) <and> (<or>(software engineering, empirical software engineering, experimental software engineering, software project management, software project)))
- ACM: ((Abstract:simulation) and (Abstract:"project management", or Abstract:"software development process", or Abstract:"software process", or Abstract:"experimental software engineering", or Abstract:"empirical software engineering")) and (PublishedAs:periodical OR PublishedAs:proceeding OR PublishedAs:thesis OR PublishedAs:report)
- Interscience: "software process simulation" AND "Project Management"; "software development" AND simulation; system dynamic AND software process OR project;simulation

Após ser conduzida a pesquisa pelas palavras-chave, sinônimos e argumentos acima, os resumos dos artigos encontrados foram lidos e realizada a seleção inicial ou exclusão dos artigos. Posteriormente, os trabalhos classificados tiveram que ser analisados na íntegra e realizada uma nova filtragem, que resultou em oitenta e nove artigos.

Estes oitenta e nove artigos selecionados foram detalhadamente estudados e as questões, a seguir, respondidas para cada um deles:

- Q1 – Quais iniciativas têm sido realizadas para a SPS?
- Q2 – Quais os benefícios da SPS?
- Q3 – Quando simular?
- Q4 – Como simular?
- Q5 – O quê simular?
- Q6 – Quais os modelos de simulação utilizados?
  - Q6.1 – Quais as características dos modelos?
  - Q6.2 – Quais as potencialidades dos modelos?
  - Q6.3 – Quais as limitações dos modelos?
- Q7 – Como é possível combinar estratégias de simulação?

O resultado deste trabalho foi contextualizado em uma tabela com as respostas para estas questões e está disponível no Anexo A deste trabalho. O desenvolvimento desta revisão possibilitou estruturar e mapear os conhecimentos sobre o domínio da simulação, sendo útil para abreviar futuros trabalhos, estabelecer relações entre eles e, principalmente, oferecer o embasamento para o desenvolvimento do modelo de simulador proposto.



O conjunto de tecnologias computacionais identificadas neste estudo e no trabalho desenvolvido por (ZHANG, KITCHENHAM e PFAHL, 2008) foi avaliado de forma comparativa, conforme descrito na sessão a seguir.

## 3.2 Avaliação comparativa dos simuladores

Os simuladores *open source* ou disponíveis para avaliação foram obtidos da revisão sistemática desenvolvida por (ZHANG, KITCHENHAM e PFAHL, 2008), da revisão sistemática descrita no capítulo anterior e também das experiências do nosso grupo de trabalho.

O desenvolvimento da avaliação comparativa teve como passo inicial a categorização dos softwares entre simuladores exclusivos para processos de software e de propósito geral.

### 3.2.1 Simuladores avaliados

Os simuladores exclusivos para processos de software identificados foram o SimSE (SIMSE, 2011), o Simulador Estocástico Baseado em Conhecimento (NICOLAU, 2009), o Charon (MURTA, BARROS e WERNER, 2002), o ProcessModel (PROCESSMODEL, 2011), o SigmaFlow (SIGMAFLOW, 2011) e o SimProcess (SIMPROCESS, 2011), destes, somente o Simulador Estocástico Baseado em Conhecimento (NICOLAU, 2009) e o Charon (MURTA, BARROS e WERNER, 2002) estão incorporados a um PSEE.

O SimSE (SIMSE, 2011) é um ambiente educativo para a simulação da engenharia de software, cujo objetivo é integrar a grande quantidade de conhecimento conceitual da engenharia de software repassada aos alunos nas aulas e a quantidade comparativamente pequena de experiência prática nesta área. O SimSE permite aos estudantes experimentar virtualmente o processo de engenharia de software (ou sub-processo) em um ambiente totalmente gráfico e interativo, que fornece *feedback* gráfico e permite compreender as causas e efeitos das complexas relações que permeiam os processos da engenharia de software.

O Simulador Estocástico Baseado em Conhecimento (NICOLAU, 2009) utiliza um mecanismo para, a partir de um histórico de execuções de processo, realizar previsões acerca de variáveis do processo de software, tais como: duração de atividades, custo e número de erros nos produtos de trabalho. Para isso, o simulador utiliza um mecanismo de raciocínio baseado em casos para calibrar o modelo de simulação em relação aos dados organizacionais e, com base nesse modelo calibrado, realiza simulações utilizando o método de Monte Carlo (NAYLOR, BALINTFY, *et al.*, 1966).

A ferramenta Charon (MURTA, BARROS e WERNER, 2002) é extensível, baseada em agentes inteligentes, fornece suporte para a modelagem, simulação, instanciação, execução, acompanhamento, monitoramento e evolução de processos de software. A arquitetura da ferramenta permite a separação entre o controle, representado por agentes, e as informações de execução do processo, representadas por bases de conhecimento. Além disso, a ferramenta pode ser estendida através da construção de novos agentes, que implementam novos requisitos. Essa construção ocorre por meio da instanciação de um *framework* e do uso de uma ontologia que define a API para a comunicação com as bases de conhecimento.

O ProcessModel (PROCESSMODEL, 2011) possibilita a captura de processos, dispõe de recursos de simulação dos modelos e gera relatórios que sugerem melhorias

na implementação dos processos. Projetado especificamente para gestores e planejadores, o ProcessModel permite inserir os principais parâmetros e informações sobre o modelo e simular o modelo. Além de capturar, permite analisar e incrementar processos cada vez mais complexos. Fornece análise replicada, definição de variáveis e atributos, e relatórios personalizados. Possibilita a criação de modelos hierárquicos a fim de melhorar a organização e o gerenciamento de grandes projetos de modelagem.

O SigmaFlow (SIGMAFLOW, 2011) permite a simulação e análise do processo com detalhes estatísticos de cada transação. Dispõe de mapas estáticos do processo que podem ser manipulados como modelos dinâmicos, cria mapas multivalorados a nível de sub-processos, possibilitando o desenvolvimento hierárquico da organização, facilitando a compreensão e a importância dos processos. Oferece funcionalidades que estimulam a criação de modelos padrão para qualquer procedimento, associando-os aos documentos de referência, ferramentas e um conjunto de métricas.

O SimProcess (SIMPROCESS, 2011) é apresentado como uma solução de apoio aos sistemas de negócios, permitindo a modelagem, simulação e análise do processo, identificando pontos críticos e disponibilizando uma perspectiva de desempenho. Desta forma, proporciona o desenvolvimento de processos qualificados e de riscos reduzidos, devido à configuração e implementação de decisões.

Na categoria dos simuladores de diferentes domínios, inclusive processos de software, enquadram-se o Vensim (VENSIM, 2011), o Extend (KRAHL, 2003), o iThink (ITHINK, 2011), o QSim (QSIM, 2011), o Netlogo (FUKS e KAWA, 2009), o Repast (NORTH, COLLIER e VOS, 2006), o DSOL (M., VERBRAECK e RENGELINK, 2005), o DEVSim++ (KIM, HAM e KIM, 1993), o DEVSimJava (KIM, SARJOUGHIAN e ELAMVAZHUTHI, 2009), o QNAP2 (VERAN e POTIER, 1984) e o SESAM (KLÜGL, HERRLER e FEHLER, 2006).

O Vensim (VENSIM, 2011) é um dos principais representantes de uma categoria de ferramentas que apoiam a simulação por sistemas dinâmicos, possibilitando integrar, documentar, simular, analisar e otimizar os modelos. Os modelos podem ser construídos graficamente ou através do editor de texto. Dentre as características, inclui funções dinâmicas, vetores, análise da sensibilidade, otimização, manuseio de dados, aplicações para interfaces, entre outras. Interligando os componentes o sistema estabelece relações entre as variáveis. Esta informação é utilizada pelo editor de equações, que permite a elaboração de um roteiro completo da simulação do modelo.

Originalmente lançado em 1988, o Extend (KRAHL, 2003) foi um dos softwares pioneiros na indústria da simulação e, a partir de sua interface gráfica, trouxe recursos que anteriormente estavam apenas disponíveis em computadores com grande poder de processamento. O ambiente de simulação Extend fornece ferramentas e modelos para os diferentes níveis de projeto. Permite criar, validar, verificar e reutilizar o modelo a ser simulado.

O simulador iThink (ITHINK, 2011) permite tomar decisões baseadas nos resultados da simulação de processos e cenários. Possibilita uma visão operacional do sistema inteiro. Ignora as metodologias lineares e fornece uma visão sobre o desempenho do sistema, detalhando as interdependências e predizendo se as metas operacionais para o sistema podem ser atingidas.

O QSim (QSIM, 2011) e suas extensões para simulação qualitativa foi desenvolvido para o raciocínio baseado em modelos com conhecimento incompleto. A representação

do QSim para equações diferenciais qualitativas e comportamentos qualitativos foi originalmente motivada por estudos baseados na análise de especialistas. A simulação qualitativa pode ser garantida para produzir um conjunto de descrições de comportamento.

O Netlogo (FUKS e KAWA, 2009) é um ambiente de modelagem programável para simular fenômenos naturais e sociais. É adequado à modelagem de sistemas complexos de desenvolvimento ao longo do tempo. Os usuários podem dar instruções a centenas ou milhares de agentes que agem de forma independente, tornando possível explorar a conexão entre o comportamento dos indivíduos, do nível micro aos padrões em nível macro que emergem da interação de muitos indivíduos.

O Repast (NORTH, COLLIER e VOS, 2006) é um kit de ferramentas livres, de código aberto, que possui um conjunto de características abstratas e três implementações concretas. Concentra-se na modelagem do comportamento social, mas não é limitado à simulação social. Foi desenvolvido na Universidade de Chicago, com a colaboração do Argonne National Laboratory. O Repast é utilizado em uma grande variedade de aplicações como os sistemas sociais, os sistemas evolutivos, a modelagem de mercado, a análise industrial, etc.

O DSOL (M., VERBRAECK e RENGELINK, 2005) é um software de código aberto baseado nas classes Java para simulação de eventos contínuos e discretos. A primeira versão do DSOL foi introduzida na IEEE's Winter Simulation Conference 2002. A idéia que motivou o desenvolvimento do DSOL baseou-se no fato da *web* fornecer as ferramentas e técnicas para especificar os conceitos da simulação em um conjunto fracamente acoplado de *web-services*.

O DEVSim++ (KIM, HAM e KIM, 1993) é um ambiente de simulação orientada a objetos que implementa módulos de Zeigler associado a um simulador abstrato em C++. Devido à modelagem orientada a objetos, o tamanho dos modelos de simulação no DEVSim++ são limitados ao tamanho da memória. A simulação no DEVSim++ mostra que esta limitação é um dos principais obstáculos para realizar simulações em grande escala. No DEVSim++ é preciso especificar os modelos básicos, dos quais os maiores são construídos e, como esses modelos são ligados entre si, de forma hierárquica.

O DEVJava (KIM, SARJOUGHIAN e ELAMVAZHUTHI, 2009) é uma ferramenta utilizada para simular modelos paralelos. Exibe toda a hierarquia do modelo de simulação utilizando componentes. Para qualquer modelo acoplado, uma ou mais mensagens percorrem simultaneamente os caminhos de acoplamento atômico. O DEVJava separa o controle de execução do simulador da visualização dos modelos e suas animações, que são apoiados pelo *simview*, um módulo que suporta interações com o usuário e controle de execução. O controle permite a execução de simulação lógica e em tempo real. Os estados de cada componente do modelo podem ser examinados individualmente quando o simulador está parado ou em pausa (ou seja, no final de um ciclo da simulação). A visualização das mensagens é permitida para todo o modelo. Além disso, a animação das mensagens não é sincronizada com a velocidade da execução da simulação.

O QNAP2 (VERAN e POTIER, 1984) é um ambiente de modelagem portátil que fornece recursos específicos para a construção, manutenção e resolução de modelos. É composto por uma coleção de algoritmos de resolução, que inclui simulação evento discreta, método matemáticos de aproximação e exatos, além de uma interface para a descrição do modelo, controle, análise e apresentação de resultados.

O SESAM (KLÜGL, HERRLER e FEHLER, 2006) é uma plataforma de modelagem e simulação multiagente. Os agentes são entidades ativas da simulação e seu comportamento é implementado com diagramas de atividade. Baseado em um grande número de componentes primitivos, um usuário é capaz de projetar uma simulação gráfica, sem saber a sintaxe de uma linguagem de programação tradicional. Baseado em um modelo de representação declarativa, explícita e programação visual, permite a implementação de modelos em nível de especificação. Otimizando a compilação permite a simulação eficiente do modelo de representação explícita. É aplicado com sucesso em diferentes áreas, como biologia, simulação de tráfego ou de logística (KLÜGL, HERRLER e FEHLER, 2006).

### 3.2.2 Cenário utilizado para a avaliação comparativa

Para que o processo de avaliação dos simuladores pudesse ser desenvolvido foi necessário elaborar um estudo de caso simplificado, que representa um caso comum de processo de software, apresentado na forma de um diagrama de atividades UML (Unified Modeling Language), conforme mostrado na Figura 2.1, e também os critérios agrupados e identificados a partir de conceitos relacionados à simulação, descritos na literatura, inclusive nos artigos avaliados na revisão sistemática e também a partir da experiência com o uso de simuladores.

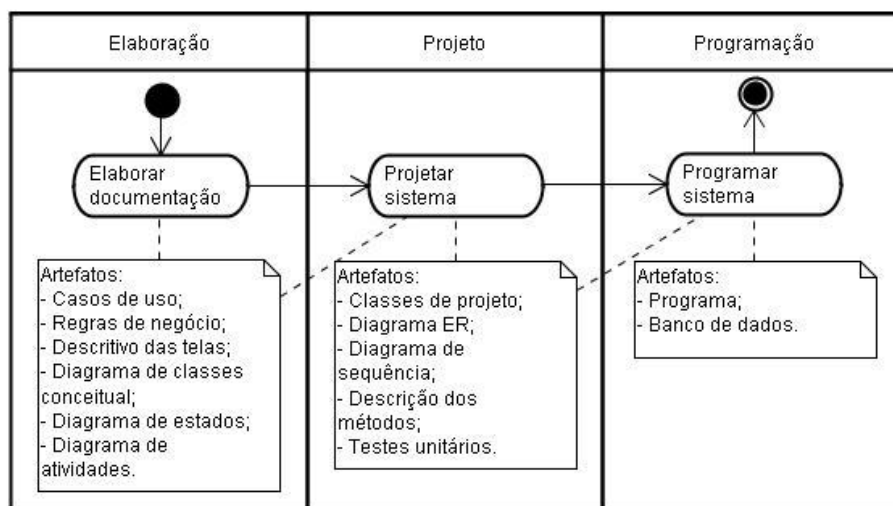


Figura 2.1: Estudo de caso simplificado utilizado no processo de avaliação.

O cenário criado é composto por três fases: elaboração, projeto e programação, que são representadas em partições. Cada fase possui uma das seguintes atividades: elaborar documentação, projetar sistema e programar sistema. Estas atividades, por sua vez, resultam em artefatos que servem de documentação para o desenvolvimento da fase seguinte, concluindo com o programa e o banco de dados. Este processo tem início com a etapa de elaboração e é finalizado com a conclusão bem sucedida da fase de programação.

### 3.2.3 Resultados da avaliação comparativa

Com a modelagem e a simulação do cenário foi possível avaliar o nível de apoio tecnológico oferecido pelas ferramentas, a partir da aplicação de um *checklist* com critérios agrupados em três contextos distintos (Modelagem, Simulação e Resultados) e contextualizados como presentes (✓), ausentes (✗) ou não foi possível avaliar (-), nas tabelas comparativas. Os critérios utilizados na avaliação comparativa, relativos à

simulação, foram obtidos dos conceitos consolidados na literatura. As características avaliadas, em relação à modelagem e aos resultados, foram obtidas a partir das experiências e anseios do nosso grupo de trabalho e do especialista.

A modelagem faz referência aos aspectos envolvidos na elaboração do cenário, conforme ilustrado na Tabela 1. Neste grupo, os critérios utilizados foram os seguintes:

- Permite criar o processo graficamente: Oferece recursos visuais para que o projetista possa escolher os objetos mais coerentes com o seu modelo, e criá-lo utilizando os recursos de clicar e arrastar;
- Permite a elaboração de equações: Dispõe de um editor de equações, associado aos processos, permitindo a execução de cálculos matemáticos, aceitando como entrada, recursos externos ou variáveis;
- Possibilita uma visualização da hierarquia de execução: Coloca à disposição do usuário a possibilidade de visualizar o processo de forma hierárquica, ou seja, pode-se visualizar sub-processos ou outras estruturas dependentes;
- Dispõe de exemplos de processos: Possui um repositório com templates que possam ser utilizados a fim de entender melhor as funcionalidades e potencialidades da ferramenta;
- Permite a reutilização de processos: Possibilita o desenvolvimento de blocos aproveitando modelos de objetos desenvolvidos em outro projeto;
- Permite importar modelos de outros programas: Possibilita a simulação de um modelo criado em outras ferramentas.

Tabela 2.1. Comparativo entre as ferramentas avaliadas, relativo à modelagem.

CARACTERÍSTICAS AVALIADAS	SimSE	Simulador Estocástico Baseado em Conhecimento	Charon	ProcessModel	SigmaFlow	SimProcess	Vensim	Extend	iThink	Qsim	Netlogo	Repast	DSOL	DEVSim++	DEVJava	QNAP2	SESAM
	Modelagem																
Permite criar o processo graficamente	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×	✓
Permite a elaboração de equações	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Possibilita a visualização da hierarquia de execução	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓
Dispõe de exemplos de processos	✓	×	×	✓	✓	✓	✓	✓	✓	×	✓	✓	-	×	×	×	✓
Permite a reutilização de processos	✓	✓	✓	✓	-	✓	-	✓	✓	-	✓	✓	×	✓	✓	×	✓
Permite importar modelos de outros programas	×	-	×	✓	×	×	-	✓	-	×	×	✓	✓	×	×	×	-

A simulação envolve as características observadas ao ser manipulado o cenário projetado, conforme ilustrado na Tabela 2. Neste grupo os critérios utilizados foram os seguintes:

- Simula um domínio específico: Questiona sobre a possibilidade da ferramenta simular qualquer problema ou é restrita a um domínio específico, como por exemplo a simulação de Processos de Software;
- Possibilita estratégias de simulação discreta: Dispõe de recursos para que a simulação ocorra com os processos situados pontualmente no tempo;
- Possibilita estratégias de simulação contínua: Dispõe de recursos para que a simulação ocorra de forma dinâmica, com alteração ao longo do processo;
- Permite a simulação de modelos determinísticos: Nestes modelos não se permite às variáveis exógenas e endógenas serem variáveis randômicas, e

as características operacionais devem ser relações exatas e não funções de densidade de probabilidade. Os modelos determinísticos são, computacionalmente, menos exigentes que os estocásticos e podem frequentemente ser resolvidos por técnicas como a do cálculo de máximos e mínimos;

- Permite a simulação de modelos estocásticos: São assim chamados quando pelo menos uma das características operacionais é dada por uma função de probabilidade. Os modelos estocásticos são também de interesse do ponto de vista da geração de amostras randômicas de dados para serem usados nos estágios de observação ou de testes de averiguação científica;
- Permite simular diferentes cenários: Armazena em um banco de dados os resultados da simulação de cada um dos cenários, com alterações drásticas (remodelagem do processo, por exemplo) ou sutis (alteração do valor de variáveis, por exemplo);
- Dispõe de recursos de animação: Potencializa o *feedback* visual utilizando técnicas de animação gráfica na simulação do modelo, ampliando a compreensão do mesmo.

Tabela 2.2. Comparativo entre as ferramentas avaliadas, relativo à simulação.

CARACTERÍSTICAS AVALIADAS	SimSE	Simulador Estocástico Baseado em Conhecimento	Charon	ProcessModel	SigmaFlow	SimProcess	Vensim	Extend	iThink	Qsim	Netlogo	Repast	DSOL	DEVSim++	DEVJava	QNAP2	SESAM
	Simulação																
Simula um domínio específico	✓	✓	✓	✓	✓	✓	×	×	×	×	×	×	×	×	×	×	×
Possibilita estratégias de simulação discreta	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Possibilita estratégias de simulação contínua	-	✓	✓	-	×	×	✓	✓	✓	✓	✓	×	✓	×	×	-	✓
Permite a simulação de modelos determinísticos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Permite a simulação de modelos estocásticos	✓	✓	-	-	-	-	✓	✓	-	-	✓	✓	×	-	×	×	✓
Permite simular diferentes cenários	✓	✓	-	✓	-	✓	✓	✓	✓	✓	✓	✓	-	✓	-	-	✓
Dispõe de recursos de animação	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×	✓

Os resultados agrupam as peculiaridades relacionadas aos dados de saída do processo de simulação, conforme ilustrado na Tabela 3. Neste grupo os critérios utilizados foram os seguintes:

- Permite armazenar os resultados obtidos: A ferramenta dispõe de um repositório de dados que armazena o histórico das simulações realizadas;
- Possibilita a personalização de relatórios: Os dados gerados a partir da simulação podem ser filtrados e organizados na forma de relatórios personalizados;
- Dispõe de recursos para que os dados sejam analisados: Disponibilidade de alternativas adicionais à personalização de relatórios;
- Identifica áreas que podem ser melhoradas: Dispõe de ferramentas de diagnóstico que identificam situações falhas no processo como gargalos, elevados custos, entre outros.



Tabela 2.3. Comparativo entre as ferramentas avaliadas, relativo aos resultados.

CARACTERÍSTICAS AVALIADAS	SimSE	Simulador Estocástico Baseado em Conhecimento	Charon	ProcessModel	SigmaFlow	SimProcess	Vensim	Extend	iThink	Qsim	Netlogo	Repast	DSOL	DEVSim++	DEVJava	QNAP2	SESAM
	Resultados																
Permite armazenar os resultados obtidos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Possibilita a personalização de relatórios	✓	-	-	✓	✓	✓	✓	✓	-	×	✓	✓	×	×	✓	✓	-
Dispõe de recursos para que os dados sejam analisados	✓	-	-	✓	✓	✓	✓	✓	✓	-	✓	✓	-	-	✓	-	✓
Identifica áreas que podem ser melhoradas	✓	×	-	✓	✓	✓	-	-	✓	✓	×	×	×	×	×	×	-

### 3.2.4 Análise da avaliação comparativa

A partir das informações obtidas e consolidadas nas três tabelas, é possível identificar o nível de apoio tecnológico oferecido por cada um dos simuladores, suas limitações e potencialidades na modelagem, simulação e análise dos resultados na gerência de processos de software.

Em relação aos critérios avaliados na modelagem, o Process Model, o Extend e o Repast apresentam todas as características avaliadas. O APSEE-Sim não permite importar modelos de outros programas por utilizar os recursos do WebAPSEE e este não apresentar esta característica. Na maioria dos simuladores (88,24%) é possível criar o processo graficamente, com recursos simples como clicar e arrastar, e também visualizar o processo, sub-processos e o fluxo de execução dos mesmos. Todas as ferramentas avaliadas possibilitam elaborar equações personalizadas para a simulação, embora algumas ofereçam este importante recurso de forma mais detalhada e avançada do que outras. A modelagem do processo não segue a mesma metodologia em todos os softwares e, neste contexto, somente em 58,82% dos casos foi verificada a disponibilidade de exemplos que pudessem facilitar esta tarefa. Os recursos para a reutilização de processos, presentes em 70,59% dos simuladores, bem como os recursos para a importação de modelos de outros programas, verificado em 23,53% dos casos, nem sempre puderam ser verificados pela ausência de documentação em diversos softwares.

A maioria dos critérios relacionados à simulação foram verificados em simuladores como o SimSE, Simulador Estocástico Baseado em Conhecimento, Vensim, Extend, Netlogo e Sesam, contudo, nenhum deles apresenta todas as características avaliadas. O APSEE-Sim não permite utilizar estratégias de simulação contínua e não dispõe de recursos de animação. A grande maioria das ferramentas (64,71%), além de simular processos de software, são de propósito geral e simulam diversas situações. Desta forma, não estão incorporadas a um ambiente de apoio ao desenvolvimento de software e não disponibilizam muitos recursos para a gerência de projetos. Em relação às estratégias de simulação, todos os softwares permitem a realização de simulações discretas, enquanto apenas 52,94% oferecem também a possibilidade de simular processos contínuos. Dados semelhantes foram verificados em relação aos modelos: todas as ferramentas permitem simular modelos determinísticos, enquanto apenas 41,18% dos softwares possibilitam a simulação de modelos estocásticos, com a utilização de equações probabilísticas; característica que em muitos casos não pode ser avaliada por limitações do software ou falta de informação sobre o recurso. Um dos propósitos da simulação é extrair novos conhecimentos a partir do processo e, para que isso ocorra é preciso armazenar os dados obtidos na simulação de diferentes cenários de forma eficiente. Os recursos desta natureza puderam ser verificados em 70,59% dos simuladores. As técnicas de apoio visual, como a animação, também puderam ser verificadas em 82,35% dos simuladores.

As características relacionadas aos resultados foram verificadas em sua totalidade no APSEE-Sim, SimSE, ProcessModel, SigmaFlow e SimProcess. Em diferentes níveis de eficiência, todos os softwares permitem armazenar os resultados obtidos, porém, em relação à análise destes dados, o resultado não foi o mesmo. Somente 58,82% dos simuladores oferecem recursos para que o gerente crie relatórios personalizados, 64,71% dispõem de funcionalidades para a análise dos dados brutos, apenas 35,29% vão além e identificam áreas que podem ser melhoradas, auxiliando na tomada de decisão nos casos considerados críticos. Contudo, assim como ocorreu na modelagem e na simulação, em muitas ferramentas os critérios não puderam ser avaliados por falta de informações ou auxílio a respeito destes recursos.

## **4 MODELO PROPOSTO E ABORDAGEM UTILIZADA PARA O DESENVOLVIMENTO DO SIMULADOR**

Este capítulo apresenta o modelo proposto para o desenvolvimento do simulador APSEE-Sim, que estende as funcionalidades do gerenciador de processos de software WebAPSEE (WEBAPSEE, 2011) com funcionalidades que permitem a condução de simulações determinísticas e estocásticas, e da análise da sensibilidade. O WebAPSEE, por ser a ferramenta escolhida para a implementação do modelo, é apresentado detalhadamente, assim como o projeto arquitetural do mecanismo de simulação, descrito utilizando a linguagem UML.

### **4.1 WebAPSEE**

O ambiente WebAPSEE é resultado de projetos de pesquisa em andamento no Laboratório de Engenharia de Software da Universidade Federal do Pará (UFPA), criado a partir da iniciativa dos professores Rodrigo Quites Reis e Carla Alessandra Lima Reis, após o retorno do Doutorado na Universidade Federal do Rio Grande do Sul (UFRGS) que originou o desenvolvimento de um PSEE denominado APSEE (LIMA, 2003). A integração das idéias e modelos gerados com o projeto APSEE, aliado à tecnologia dos Web Services, resultou no ambiente WebAPSEE.

O desenvolvimento do WebAPSEE partiu de uma modelagem específica do ciclo de vida de processos de software, com ênfase na execução do processo. Esta perspectiva permitiu a definição dos componentes do modelo deste ambiente e que foram fundamentais para o desenvolvimento do PSEE.

A Figura 4.1, extraída de (LIMA, 2003), ilustra o ciclo de vida de processos de software com as suas fases. A provisão de tecnologia inclui o fornecimento de tecnologia de suporte à produção de software e de modelos de processo (como as linguagens de modelagem de processo, modelos de processo prontos para reutilização e ferramentas para aquisição, modelagem, análise, projeto, simulação, evolução, execução e monitoração de modelos de processo).

A análise de requisitos do processo identifica os requisitos para o projeto de um novo processo, ou novos requisitos para um processo existente. Os requisitos resultantes especificam os recursos e propriedades que o processo deve oferecer.

O projeto do processo provê a arquitetura geral e detalhada do processo. Nesta etapa, as linguagens de modelagem do processo são utilizadas. A instanciação do processo modifica a especificação do processo produzida pela atividade anterior acrescentando informações detalhadas sobre os prazos, agentes e recursos utilizados por cada atividade definida no processo.

A simulação do processo permite verificação e validação dos processos definidos antes da execução. A execução do modelo de processo utiliza o processo instanciado e o executa através da invocação de ferramentas para guiar e assistir a realização do processo no mundo real. Informações sobre o andamento do processo (*feedback*) são coletadas e analisadas durante a execução.

A avaliação do processo provê informação quantitativa e qualitativa que descreve o desempenho de todo o processo em execução. A avaliação pode ocorrer em paralelo com a execução do modelo de processo e as informações adquiridas podem ser utilizadas nas futuras ocorrências da atividade de análise de requisitos.

A execução de processos de software é uma fase do ciclo de vida de processos de software na qual as atividades modeladas são realizadas tanto pelos desenvolvedores (quando demandam agentes humanos) quanto automaticamente (quando demandam a invocação de ferramentas autônomas).

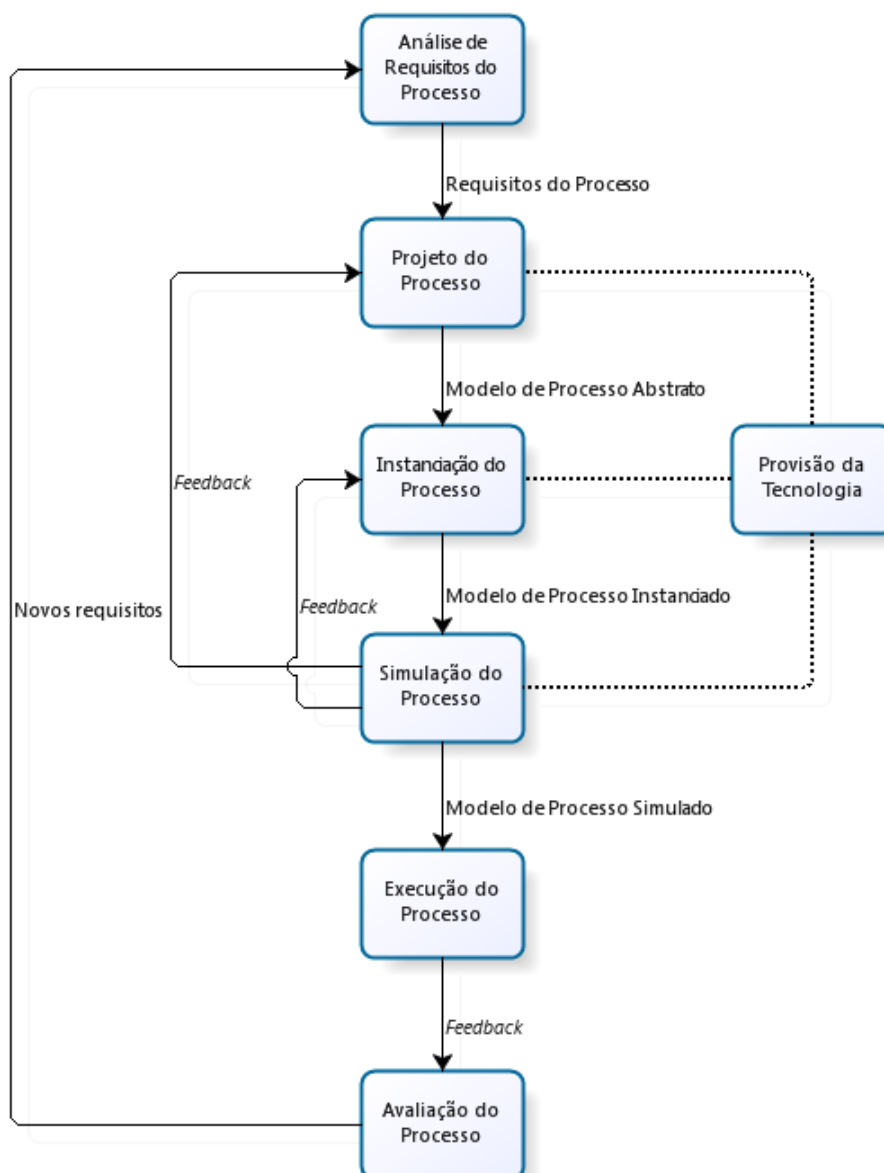


Figura 4.1: Ciclo de vida de processos de software (LIMA, 2003).

No meta-processo é fundamental a participação de agentes humanos que operem na fase de execução do processo. Além disso, é necessário um projetista de processo, que é o responsável por descrever o processo a ser executado, e um gerente do processo que deverá acompanhar a execução e a avaliação do processo, analisando seu desempenho. Em geral, os PSEEs estabelecem seu meta-processo através da provisão de tecnologia e paradigmas adotados para modelagem e execução de processos.

Apesar do forte relacionamento dos componentes envolvidos com a gerência de processos, (LIMA, 2003) identificou algumas camadas relevantes que agrupam componentes e ferramentas que compõem modelo WebAPSEE e que são utilizadas para definir sua estrutura, propiciando uma visão geral do seu funcionamento.

A separação em três camadas principais: meta-modelo, mecanismos de gerência de processos e nível de interação com o usuário, é ilustrada na Figura 4.2, adaptada de (LIMA, 2003).

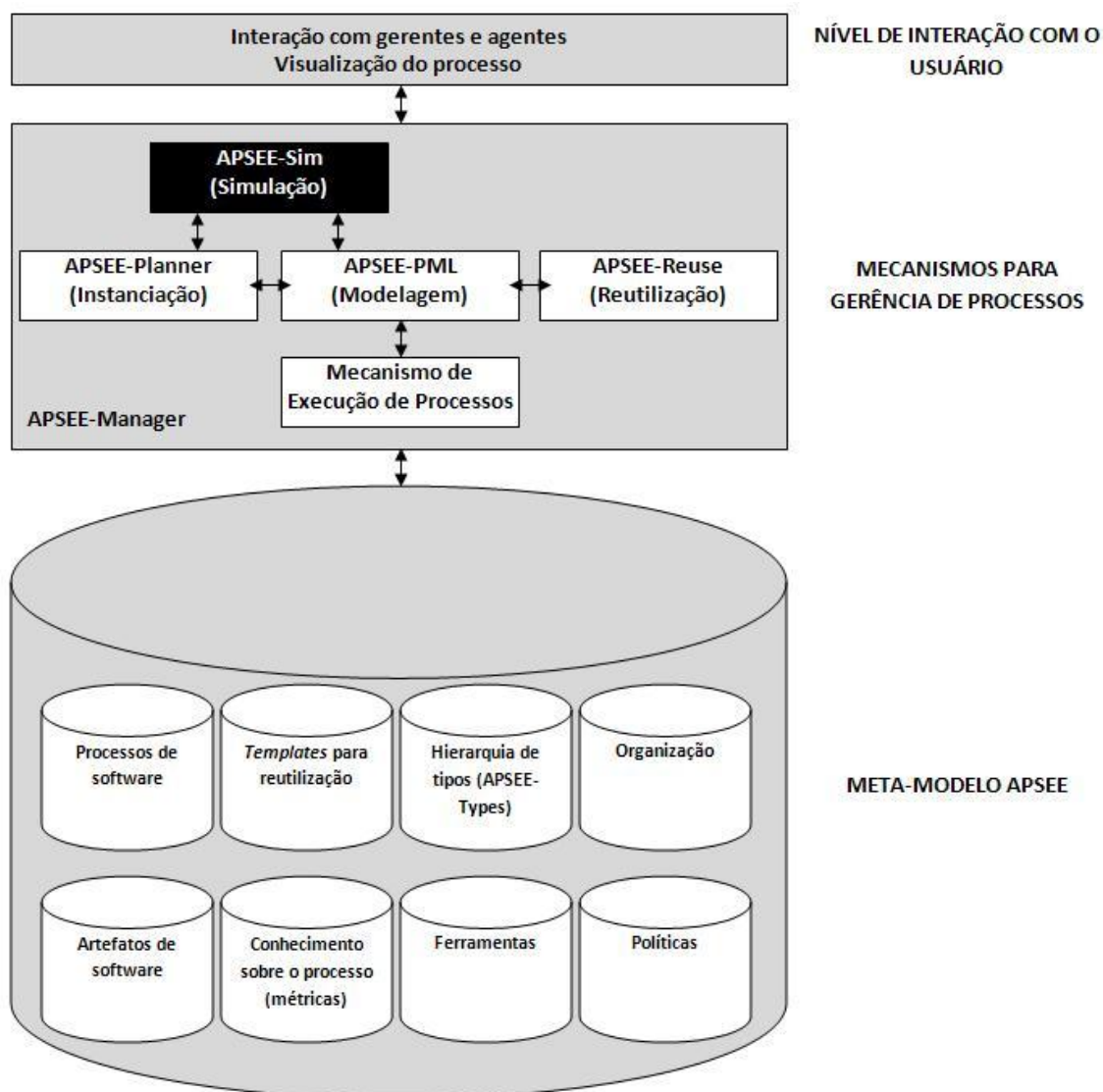


Figura 4.2: Visão geral do modelo WebAPSEE (LIMA, 2003).

O meta-modelo WebAPSEE foi proposto para facilitar a integração de diferentes serviços de gerência de processos de software, que estão na camada central, e possui os seguintes componentes:

- Processos de Software: Modelos de processo de software em seus diferentes estados (modelados em nível abstrato ou instanciado, e em diferentes estados de execução);
- Templates para reutilização: descrevem processos abstratos que podem ser reutilizados em diferentes contextos;
- Hierarquias de Tipos (APSEE-Types): relacionadas aos componentes do WebAPSEE e utilizadas na descrição de processos abstratos e de reutilização, além de permitir o raciocínio sobre elementos do processo de forma genérica;
- Organização: Incorpora o modelo de recursos de apoio utilizados pelas atividades, o modelo de pessoas da organização (agentes), suas habilidades, afinidades, cargos e grupos de trabalho;
- Artefatos de software: Correspondem aos itens de dados manipulados, criados e utilizados durante o desenvolvimento de software;
- Ferramentas: Informação sobre as ferramentas disponíveis no ambiente para realização das atividades;
- Políticas: São regras definidas pelo usuário que permitem estabelecer quando um modelo de processos está correto (estáticas), como atividades devem ser instanciadas (de instanciação) e que ações realizar na ocorrência de eventos durante execução (dinâmicas). Políticas podem ser habilitadas na organização, em processos ou atividades específicas;
- Conhecimento sobre o Processo: É o componente que permite definir e armazenar métricas e estimativas para os componentes do processo, as quais podem ser consultadas dinamicamente durante a execução do mesmo;

Os mecanismos de gerência de processos são compostos pela linguagem de modelagem e o mecanismo de execução, que interagem com todos os outros componentes e são descritos a seguir:

- APSEE-PML: Linguagem para modelagem de processos que permite definição de instâncias de processos de software e seus relacionamentos com os outros componentes do modelo;
- Mecanismo de Execução do Processo: Coordena as atividades do processo em execução através da interpretação da linguagem de modelagem de processos APSEE-PML;
- APSEE-Planner: Auxilia na instanciação de processos de software através do uso de Políticas de Instanciação definidas pelo usuário;
- APSEE-Reuse: Componente de apoio à reutilização de processos responsável pela criação, recuperação e adaptação de templates reutilizáveis;

O componente de interação e visualização do processo provê mecanismos de interação especializados para os diferentes usuários do ambiente. Por exemplo, agendas para os agentes, propostas neste trabalho, e facilidades de visualização de processos para projetista e gerentes.

A seguir, é detalhado o modelo proposto para estender as funcionalidades do WebAPSEE ou outro PSEE, possibilitando a simulação dos processos modelados.

## 4.2 Modelo proposto

Como pode ser observado na Figura 4.1, a simulação faz parte do ciclo de vida de processos de software, contudo suas funcionalidades não estão presentes na maioria dos PSEEs e também não foram originalmente implementadas no WebAPSEE. Esta carência torna mais árduo o trabalho do gerente em estimar custos, prazos e outros fatores inerentes ao desenvolvimento de processos de software.

Para que o WebAPSEE pudesse vir a dispor de recursos de simulação foi necessário alterar o seu modelo geral, conforme ilustrado na Figura 4.2, fazendo-se necessário o acréscimo de um novo mecanismo na gerência de processos, o APSEE-Sim. Este interage diretamente com o mecanismo de modelagem APSEE-PML e com o mecanismo de instânciação APSEE-Planner.

O APSEE-Sim representa o modelo de simulador (conceito) proposto neste trabalho, definido de forma genérica, em alto nível de abstração, para que possa vir a ser utilizado na implementação de simuladores em diferentes PSEEs.

Os elementos que compõem o modelo e como estes se complementam podem ser visualizados na Figura 4.3. Esta definição teve como base os estudos bibliográficos realizados no desenvolvimento da revisão sistemática e a avaliação comparativa entre os simuladores, detalhados no capítulo 3, além das principais contribuições originadas do conhecimento do especialista (gerente), obtidas por meio de entrevistas, reuniões e troca de experiências.

Como o simulador baseia-se nas informações de um processo já definido no PSEE, caso a condução de uma simulação não seja desejada, o processo pode ser executado diretamente após a modelagem. Desta forma, a etapa de modelagem do processo fica fora do escopo da simulação.

No primeiro passo da simulação ocorre a captura das informações do processo (variáveis). Nesta etapa, a base de dados do PSEE é acessada e as informações do processo extraídas, atribuindo valores às variáveis e disponibilizando-as para o desenvolvimento das equações.

Para a condução das simulações, o gerente é obrigado a atribuir equações às atividades e/ou processo, utilizando constantes, operadores, funções, variáveis que representam informações extraídas da base de dados do PSEE e/ou variáveis novas. Para que este recurso possa ser utilizado no componente implementado, é necessário a inserção da equação no próprio código fonte. Na etapa seguinte, o usuário deverá optar entre a condução de uma simulação determinística ou estocástica.

Ao escolher a realização de uma simulação determinística, o gerente poderá manipular o valor das variáveis das equações, definindo valores iguais para os valores máximos e mínimos no arquivo de propriedades. Desta forma, é possível testarem-se diferentes valores para as variáveis.

Caso escolha realizar uma simulação estocástica, será permitido manipular o valor das variáveis das equações, definindo valores diferentes para os valores máximos e mínimos no arquivo de propriedades. Neste caso, o usuário poderá gerar dados aleatórios, utilizando a técnica Monte Carlo (NAYLOR, BALINTFY, *et al.*, 1966), definindo o número de iterações desejadas.

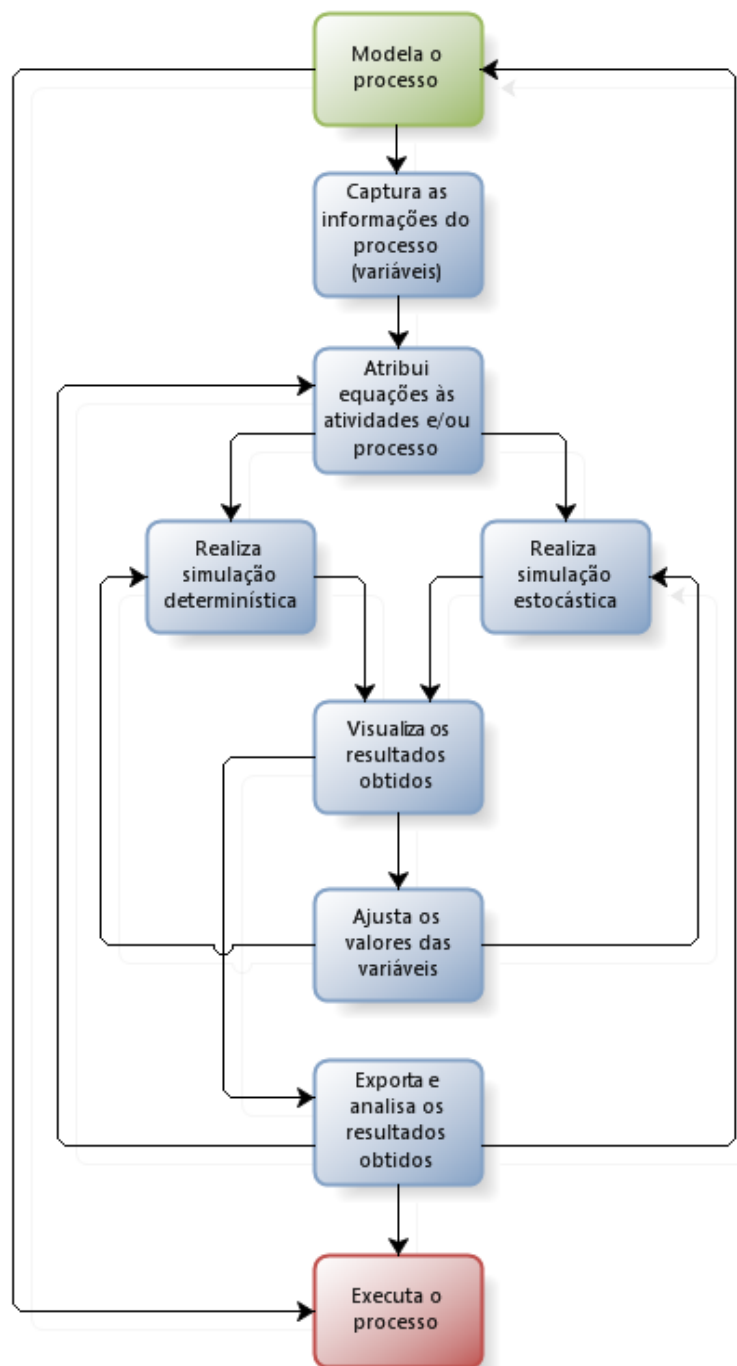


Figura 4.3. Modelo proposto para o simulador.

Ao concluir a simulação determinística e/ou estocástica, será possível visualizar os resultados obtidos com o impacto das variações de valores no resultado final (análise da sensibilidade). Em seguida, o gerente poderá ajustar os valores das variáveis ou exportar e analisar os resultados obtidos.

Se o gerente optar por ajustar os valores das variáveis, será possível executar uma nova simulação determinística ou estocástica.

Ao concluir a(s) simulação(ões), é permitindo ao gerente exportar e analisar os resultados obtidos. Nesta fase, os resultados da simulação de um (determinística) ou vários (estocástica) cenários são exportados para uma planilha eletrônica, para que seja



possível analisá-los com os diversos recursos disponíveis nestas ferramentas: gráficos, tabelas dinâmicas, filtros, etc. A partir da planilha eletrônica é possível armazenar os resultados da simulação. Assim como a modelagem do processo, a execução também não está no escopo da simulação, mas pode ser conduzida levando em consideração os resultados obtidos na simulação.

Para que este modelo pudesse ser implementado junto ao WebAPSEE, foi necessário desenvolver um projeto capaz de representar o modelo proposto. Esta prova do conceito é detalhada na sessão seguinte.

### 4.3 Implementação do APSEE- Sim

Com a definição do novo componente (APSEE-Sim) no modelo geral do WebAPSEE, ilustrado na Figura 4.2, foi necessário elaborar os casos de uso que representam os requisitos funcionais do simulador, conforme a Figura 4.4.

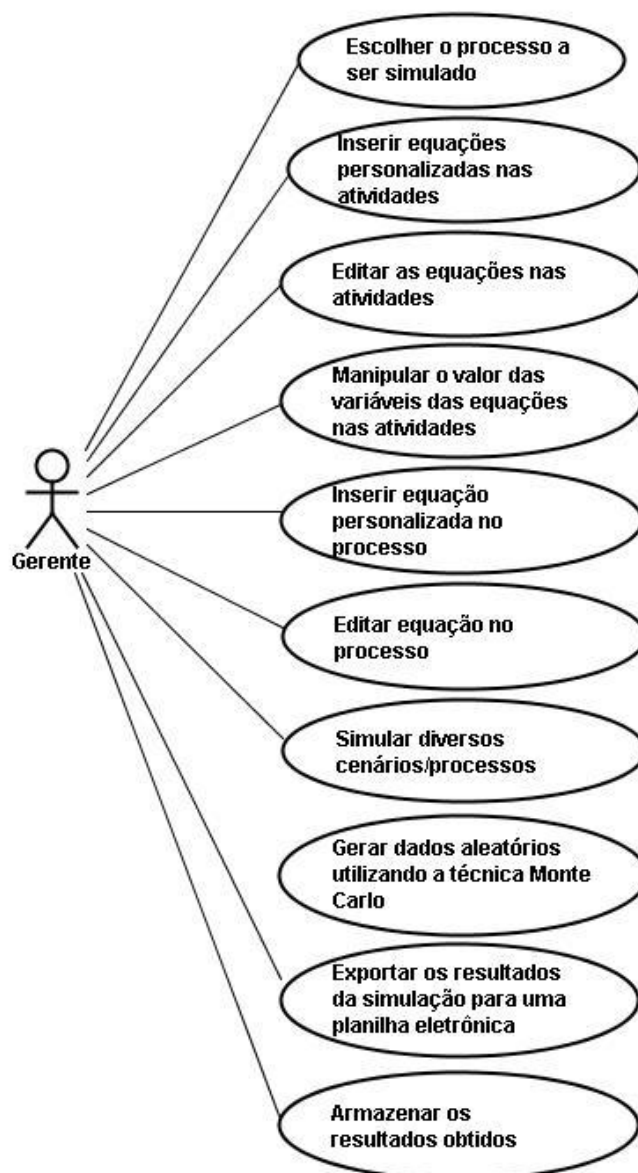


Figura 4.4: Casos de uso do simulador.

Para que o simulador possa ser utilizado de forma eficiente, ele deverá oferecer ao gerente as seguintes funcionalidades:

- Escolher o processo a ser simulado: Permitir que o gerente simule qualquer processo modelado no WebAPSEE. Esta funcionalidade está disponível na própria interface do *browser*;
- Inserir equações personalizadas nas atividades: Elaborar equações para as atividades utilizando constantes, operadores, funções, variáveis que representam informações extraídas da base de dados do WebAPSEE e/ou variáveis novas. Para que este recurso possa ser utilizado no APSEE-Sim é necessária a inserção no próprio código fonte;
- Editar as equações nas atividades: Modificar as equações existentes, acrescentar ou excluir constantes, operadores, funções, variáveis existentes ou novas. Assim como a inserção de equações, a edição só pode ser feita no código fonte;
- Manipular o valor das variáveis das equações nas atividades: Permitir que o gerente teste diferentes valores para as variáveis utilizadas nas equações e, desta forma, consiga elaborar diversos cenários e visualizar, de forma prática, o impacto destas variações de valores no resultado final (análise da sensibilidade). Para que diferentes valores possam ser aplicados às variáveis é necessário alterá-los no arquivo de propriedades;
- Inserir equação personalizada no processo: Além do gerente poder utilizar uma equação para cada atividade, é possível que ele elabore uma equação para o processo utilizando também as constantes, operadores, funções, variáveis que representam o resultado das equações atribuídas às atividades e/ou variáveis novas. Para que este recurso possa ser utilizado no APSEE-Sim é necessária a inserção no próprio código fonte;
- Editar equação no processo: A equação atribuída ao processo também pode ser alterada, acrescentando ou excluindo constantes, operadores, funções, variáveis existentes ou novas. Assim como a inserção de equações, a edição só pode ser feita no código fonte;
- Simular diversos cenários/processos: Manipulando o valor das variáveis o gerente poderá elaborar diversos cenários e os resultados obtidos poderão ser exportados para uma planilha eletrônica;
- Gerar dados aleatórios utilizando a técnica Monte Carlo: Caso o gerente necessite uma vasta base de dados para a análise, fica inviável armazenar manualmente cada um dos cenários. Neste caso o usuário poderá gerar dados aleatórios, utilizando a técnica Monte Carlo, escolhendo valores máximos e mínimos para as variáveis e definindo o número de iterações desejadas (simulação estocástica), também no arquivo de propriedades;
- Exportar os resultados da simulação para uma planilha eletrônica: Após a simulação de um (determinística) ou vários (estocástica) cenários o gerente poderá exportar os resultados para uma planilha eletrônica, para que possa analisar os dados com os diversos recursos disponíveis nestas ferramentas (gráficos, tabelas dinâmicas, filtros, etc.);
- Armazenar os resultados obtidos: A partir da planilha eletrônica é possível armazenar os resultados da simulação, que poderá ser incrementado, sempre que for necessário.

Além da definição dos casos de uso, foi necessário fazer ajustes no modelo de dados do WebAPSEE para representar o modelo conceitual do simulador. O projeto do banco de dados pode ser visualizado na Figura 4.5, na forma de um diagrama Entidade Relacionamento.

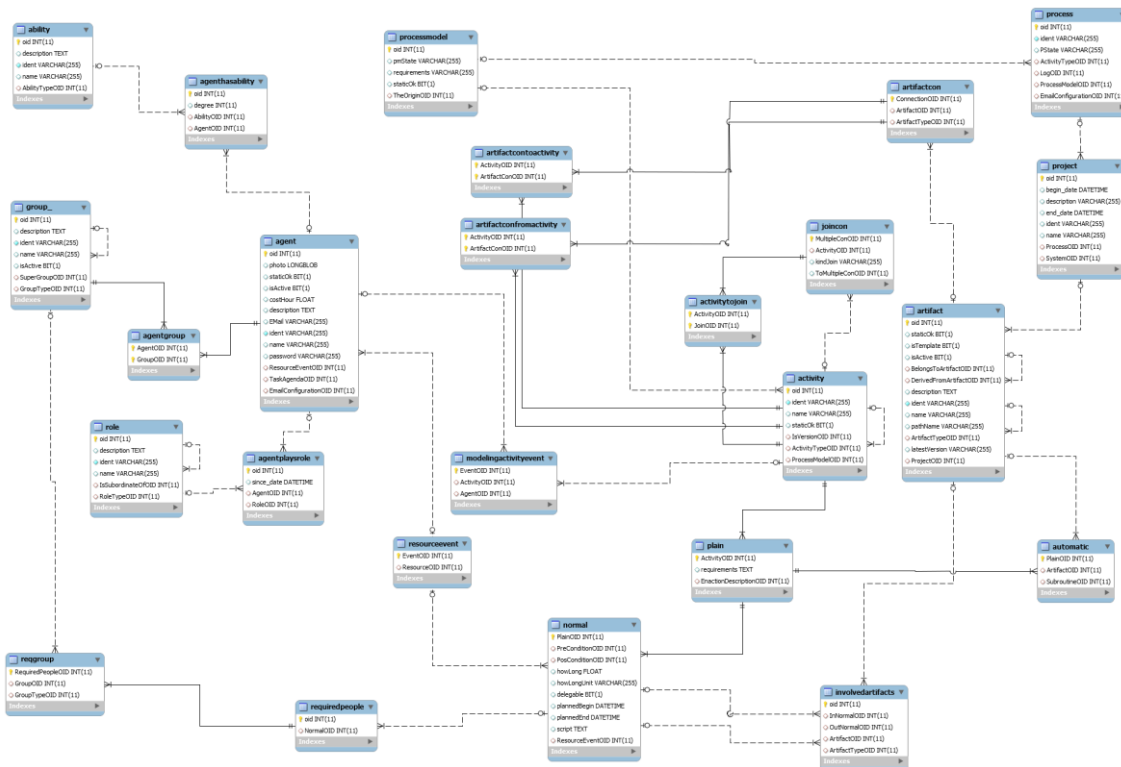


Figura 4.5: Diagrama entidade relacionamento do simulador.

O fluxo de trabalho de um caso de uso descreve o que o protótipo deve fazer para fornecer o resultado que o ator necessita. A estrutura do fluxo de trabalho do simulador foi descrita graficamente com a ajuda de um diagrama de atividades, conforme ilustrado na Figura 4.6. Este diagrama detalha a sequência das atividades para atender a cada caso de uso e permitir ao gerente atingir os objetivos desejados.

Para que estas funcionalidades pudessem ser oferecidas, foi preciso implementar os seguintes recursos:

- Permitir que o gerente acesse o simulador a partir do navegador, que por sua vez, se conecta ao banco de dados do WebAPSEE e retorna as informações sobre os processos;
- Possibilitar escolher o processo a ser simulado, dentre os vários processos modelados no WebAPSEE;
- Dispor de opções para a visualização dos resultados de uma simulação determinística (*Simulation Details*) ou estocástica (Monte Carlo);
- Oferecer a opção de exportar os resultados da simulação para um arquivo compatível com as planilhas eletrônicas, que oferecem diversos recursos para que estes dados sejam manipulados, utilizando a análise da sensibilidade;
- Permitir criar ou modificar as equações desejadas (tempo, custo, qualidade ou outra), utilizando constantes, operadores, funções e variáveis para a(s) atividade(s) e/ou processo(s);

- Possibilitar ajustar as propriedades relacionadas à simulação determinística e estocástica (Monte Carlo), testando diferentes valores para as variáveis utilizadas na equação;
- Executar a quantidade desejada de simulações, nos processos desejados.

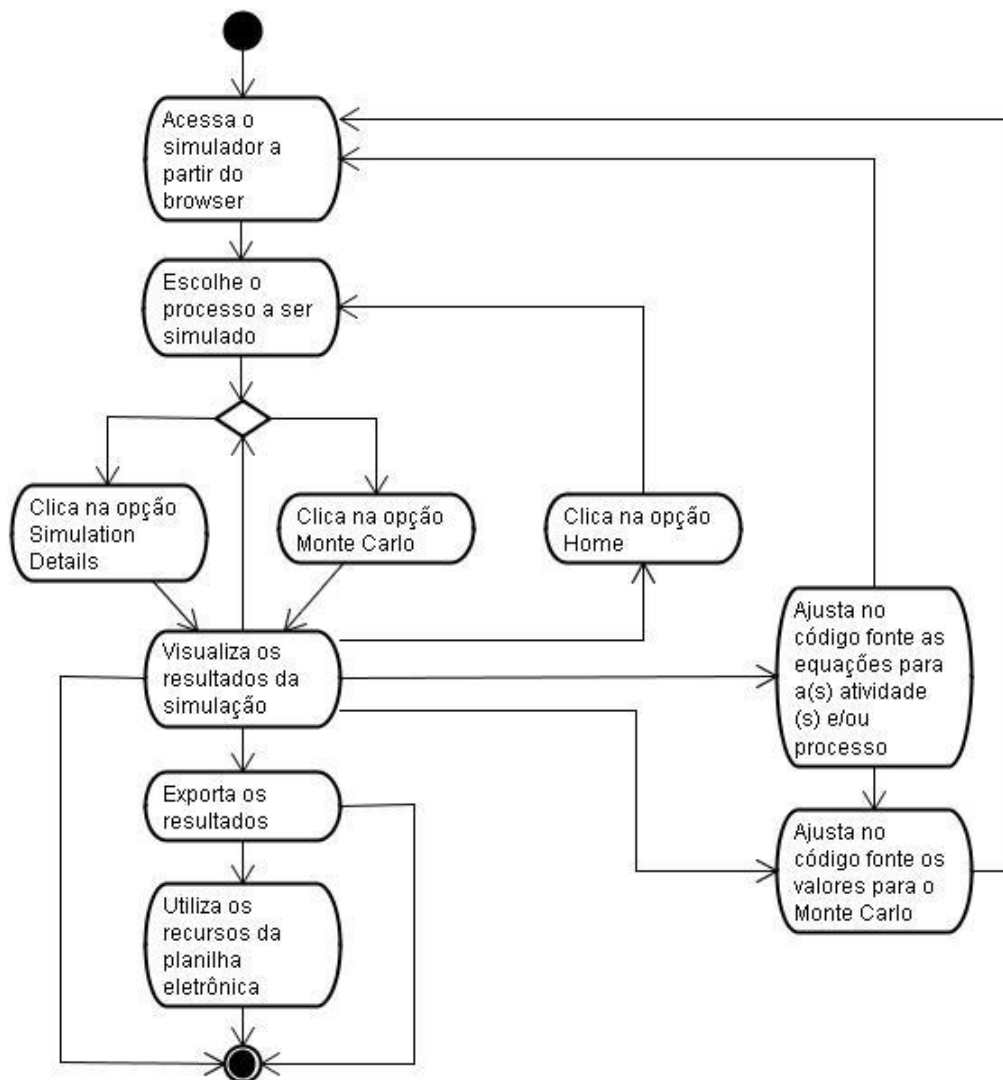


Figura 4.6: Diagrama de atividades do simulador.

A representação da estrutura e relações das classes do simulador, que servem de modelo para os objetos, é expressa através do diagrama da Figura 4.7, que contém o nome, os atributos e os métodos das classes. A descrição detalhada destas classes encontra-se no Anexo B deste documento.

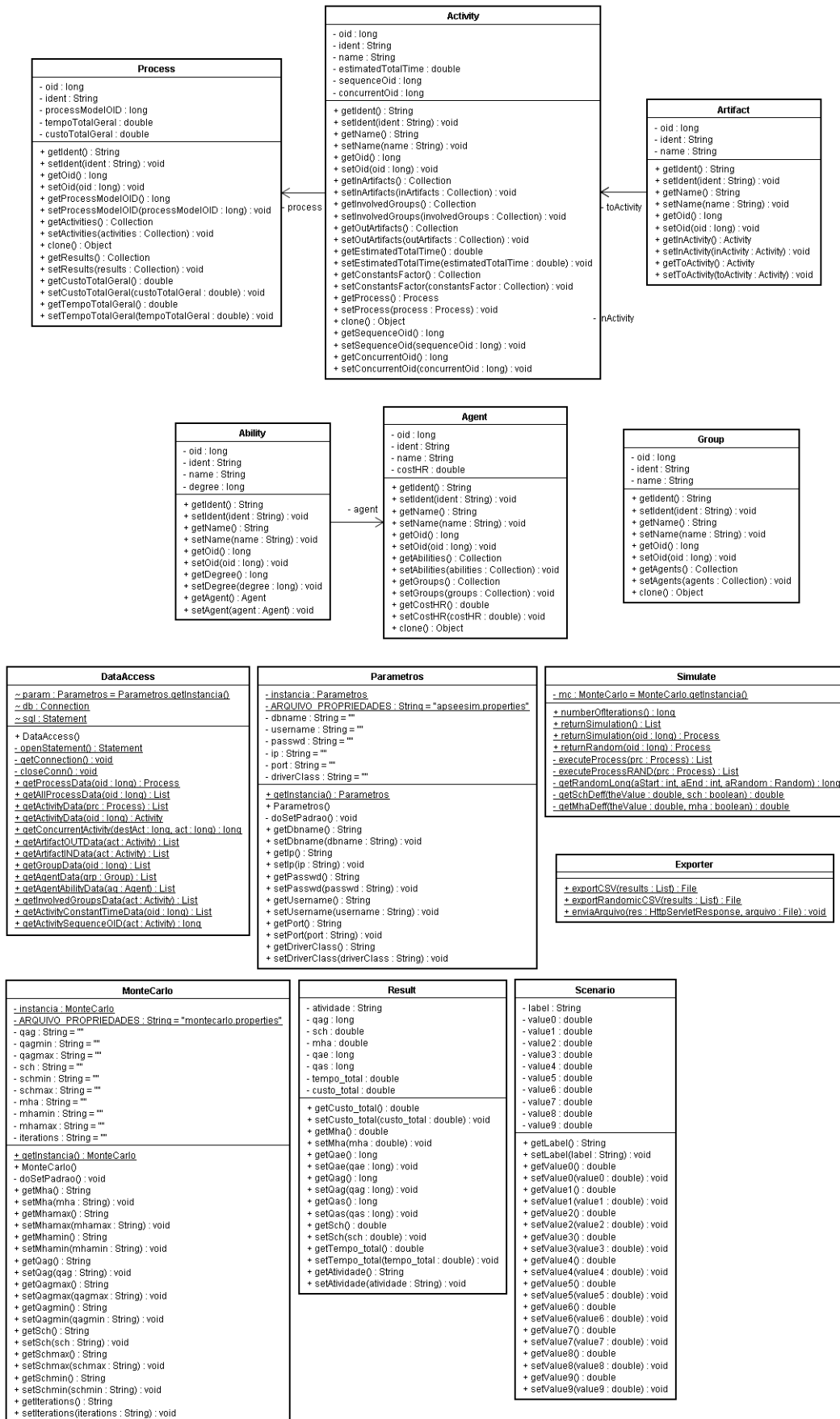


Figura 4.7: Diagrama de classes do simulador

O desenvolvimento do projeto resultou na interface *web* representada na Figura 4.8, disponibilizando ao gerente os recursos desejados para a condução das simulações que utilizam os processos modelados no WebAPSEE e exportam os resultados para uma planilha eletrônica.

Home      Simulation Details      Monte Carlo							
-* Process Simulation Detail *-							
This simulation results							
Atividade	Qt. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
Estudo de caso.Elaborar os Diagramas Comportamentais	3	R\$ 180,00	5,33	0	2	0,00	R\$ 2.400,00
Estudo de caso.Elaborar os Serviços	3	R\$ 175,00	5,00	0	2	9,00	R\$ 12.600,00
Estudo de caso.Elaborar a Interface	4	R\$ 215,00	5,25	0	2	10,00	R\$ 17.200,00
Estudo de caso.Projetar o Diagrama de Interação	3	R\$ 180,00	5,33	0	1	0,00	R\$ 2.000,00
Estudo de caso.Projetar os Diagramas Estruturais	4	R\$ 220,00	4,75	0	4	19,50	R\$ 34.320,00
Estudo de caso.Projetar os Testes	3	R\$ 180,00	5,33	0	1	1,00	R\$ 1.440,00
Estudo de caso.Implementar	13	R\$ 540,00	4,54	0	3	13,00	R\$ 56.160,00
<b>T O T A L</b>						<b>52,50</b>	<b>R\$ 126.120,00</b>

Figura 4.8: Interface do APSEE-Sim

O capítulo a seguir detalha o estudo de caso utilizado para a condução das simulações determinísticas e estocásticas, os resultados obtidos e a avaliação do simulador de acordo com os critérios utilizados no capítulo anterior.

## 5 ESTUDO DE CASO, RESULTADOS E AVALIAÇÃO DO SIMULADOR

Para que o simulador APSEE-Sim, desenvolvido a partir do modelo proposto, pudesse realizar simulações determinísticas e estocásticas, gerando resultados para a condução de diversas análises, inclusive da sensibilidade e, desta forma, ter seus recursos avaliados, foi necessário adotar um estudo de caso com informações detalhadas para as atividades e para o processo com um todo. O estudo de caso modelado é detalhado abaixo. Em seguida, são apresentados os resultados obtidos na condução de simulações determinísticas e estocásticas, além da análise destes dados, inclusive da sensibilidade. A utilização do simulador também permitiu avaliá-lo de acordo com os critérios utilizados no capítulo 3 deste trabalho.

### 5.1 Detalhamento do Estudo de caso

O estudo de caso utilizado neste trabalho foi baseado em um caso real de duas empresas que trabalham de forma colaborativa no desenvolvimento de software. O processo foi modelado no WebAPSEE, conforme a Figura 5.1, e compreende as atividades:

- Elaborar os diagramas comportamentais;
- Elaborar os serviços;
- Elaborar a interface;
- Projetar o diagrama de interação;
- Projetar os diagramas estruturais;
- Projetar os testes e;
- Implementar.

Cada uma destas atividades é desenvolvida por um grupo (empresa), que agrupa os agentes (funcionários) e, tem como objetivo produzir os artefatos necessários para o desenvolvimento da atividade seguinte, documentação e/ou resultar no produto final.

A atividade de **Elaborar os diagramas comportamentais** é desenvolvida pelo grupo Diagramas Comportamentais, que reúne os seguintes agentes: Coordenador Geral, Gerente Cx e Analista Cx1. No WebAPSEE também foram cadastrados os detalhes de todos os agentes, como: Custo por hora, Habilidades, entre outros. Esta atividade tem como objetivo produzir os artefatos Regras de negócio e Caso de uso.

Em paralelo à atividade Elaborar os diagramas comportamentais ocorre a atividade **Elaborar serviços**, desenvolvida pelo grupo Serviços, compreendendo os agentes: Coordenador Geral, Gerente Cx e Analista Cx2. Os artefatos produzidos nesta atividade são Casos de teste e *web-services*.

No instante que em que são concluídas as duas atividades citadas acima, é iniciada a atividade **Elaborar interface** pelo grupo Interface, que é composto pelos agentes: Coordenador Geral, Gerente Cx, Analista Cx1 e Prototipador Cx. Como resultado são desenvolvidos os artefatos Descritivo de telas e Protótipo das telas.

As três atividades anteriores são desenvolvidas pela Empresa 1 e as atividades seguintes pela Empresa 2. As atividades **Projetar o diagrama de interação** e **Projetar os diagramas estruturais** são desempenhadas em paralelo. Projetar o diagrama de interação é desenvolvida pelo grupo Diagrama de Interação, que é composto pelos agentes: Coordenador Geral, Gerente Lj e Analista Sup. Esta atividade produz como artefato o Diagrama de seqüências.

**Projetar os diagramas estruturais** é desenvolvida pelo grupo Diagramas Estruturais, que reúne os seguintes agentes: Coordenador Geral, Gerente Lj, Analista Ctb e Analista Edu. Esta atividade tem como objetivo produzir os seguintes artefatos: Diagrama de classes, Diagrama de atividades, Descritivo dos métodos e Diagrama Entidade Relacionamento.

A atividade **Projetar testes** é iniciada no momento em que as atividades Projetar o diagrama de interação e Projetar os diagramas estruturais são concluídas. Esta atividade é desenvolvida pelo grupo Testes, que compreende os seguintes agentes: Coordenador Geral, Gerente Lj e Analista Sup. Como resultado desta atividade é desenvolvido o artefato Testes unitários.

A última atividade, **Implementar**, é iniciada quando a atividade Projetar testes for concluída, que compreende os detalhes dos agentes: Coordenador Geral, Gerente Lj, Analista Ctb, Analista Edu, Analista Fin, Analista Sup, Programador Ctb1, Programador Ctb2, Programador Edu1, Programador Edu2, Programador Fin1, Programador Fin2 e Programador Sup1. Este grande grupo tem como objetivo desenvolver o produto final, ou seja, os artefatos Programa, Banco de dados e Testes unitários.

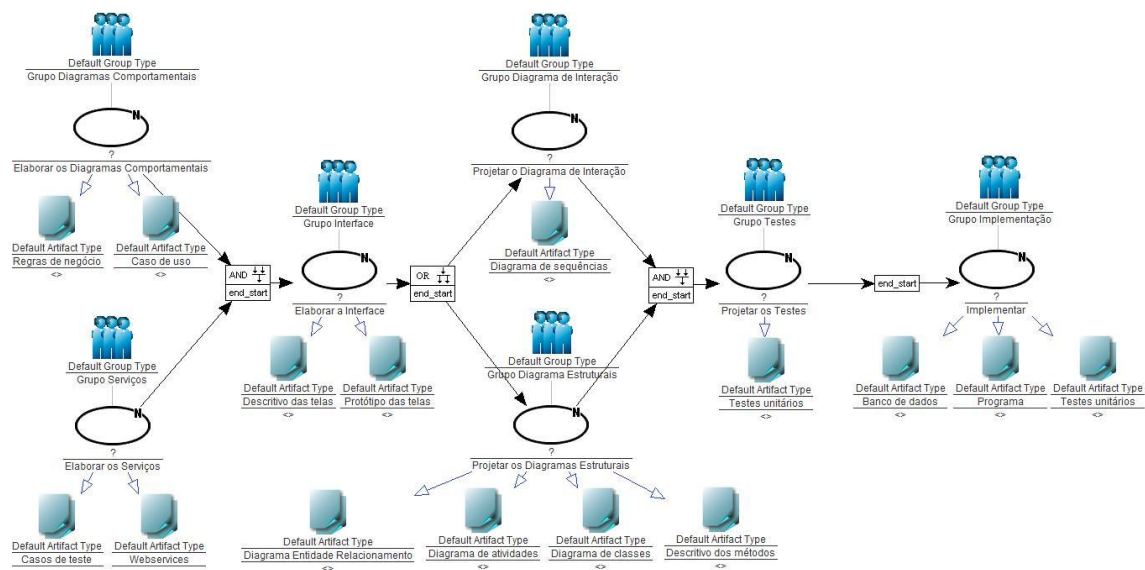


Figura 5.1: Estudo de caso modelado no WebAPSEE

Considerando a vasta experiência do coordenador geral no desenvolvimento de projetos de desenvolvimento de software, como por exemplo o estudo de caso utilizado neste trabalho, foi possível obter, através de documentos, reuniões e entrevistas, as informações necessárias para a elaboração das equações com o cálculo aproximado do



tempo e custo necessários para o desenvolvimento de cada uma das atividades e, consequentemente, do processo como um todo.

No WebAPSEE, as equações podem ser elaboradas considerando as seguintes variáveis, que representam os detalhes modelados para os agentes, grupos, recursos, artefatos e para as atividades normais do processo:

- Agente:
  - Custo por hora (CPH);
  - Quantidade de papéis (QDP);
  - Quantidade de afinidades (QDA);
  - Média do nível de afinidade (1-5) (MNA);
  - Quantidade de habilidades (QDH);
  - Média do grau de habilidades (1-6) (MGH);
  - Quantidade de grupos ao(s) qual(is) pertence(m) (QGP).
- Grupo(s):
  - Quantidade de agentes (QAG);
  - Média do custo por hora (MCH);
  - Quantidade de papéis (QPA);
  - Quantidade de afinidades (QAF);
  - Média do nível de afinidade (1-5) (MAF);
  - Quantidade de habilidades (QHA);
  - Média do grau de habilidades (1-6) (MHA);
  - Quantidade de grupos ao(s) qual(is) pertence(m) (QPR).
- Recurso(s):
  - Quantidade de recursos consumíveis (QRC);
  - Quantidade de recursos exclusivos (QRE);
  - Quantidade de recursos compartilhados (QCO);
  - Quantidade de recursos ao(s) qual(is) pertence(m) os recursos consumíveis (QCN);
  - Quantidade de recursos ao(s) qual(is) pertence(m) os recursos exclusivos (QEX);
  - Quantidade de recursos ao(s) qual(is) pertence(m) os recursos compartilhados (QCM);
  - Média do tempo entre falha dos recursos consumíveis (MFN);
  - Média do tempo entre falha dos recursos exclusivos (MFE);
  - Média do tempo entre falha dos recursos compartilhados (MFM);
  - Média do custo dos recursos consumíveis (MCN);
  - Média do custo dos recursos exclusivos (MCE);
  - Média do custo dos recursos compartilhados (MCM);
  - Quantidade de recursos requeridos pelos recursos consumíveis (QRN);
  - Quantidade de recursos requeridos pelos recursos exclusivos (QRX);
  - Quantidade de recursos requeridos pelos recursos compartilhados (QRM);
- Artefato(s):
  - Quantidade de artefatos de entrada (QAE);
  - Quantidade de artefatos de saída (QAS);
  - Quantidade de artefatos derivados dos artefatos de entrada (QDE);
  - Quantidade de artefatos derivados dos artefatos de saída (QDS);

- Quantidade de artefatos pertencentes aos artefatos de entrada (QPE);
- Quantidade de artefatos pertencentes aos artefatos de saída (QPS).
- Atividade:
  - Duração em dias (DED);
  - Atividade delegável (ATD);
  - Data de início planejado (DIP);
  - Data de término planejado (DTP);
  - Estimativa em horas (EEH);
  - Data de início realizado (DIR);
  - Tipo de atividade (TDA);
  - Quantidade de versões (QDV);
  - Quantidade de agentes requeridos (QAR);
  - Quantidade de grupos requeridos (QGR).

Além das variáveis mencionadas, o gerente poderá criar novas, utilizar constantes, operadores e funções disponíveis na linguagem de programação.

- Constantes:
  - Conjunto dos números naturais, que podem ser formados pelos algarismos de 0 a 9.
- Operadores:
  - Adição (+);
  - Subtração (-);
  - Multiplicação (\*);
  - Divisão (/);
  - Decimal (,);
  - Exponencial (^);
  - Porcentagem (%);
  - Igual (=);
  - Diferente (<>);
  - Maior (>);
  - Maior ou igual (>=);
  - Menor (<);
  - Menor ou igual (<=).
  - AND;
  - OR;
  - NOT e;
  - Parênteses ( ( ) ).
- Funções:
  - ABS;
  - DELAY;
  - EXP;
  - IF THEN ELSE;
  - INTEGER;
  - LN;
  - MAX;
  - MIN;
  - MODULO;
  - PULSE;
  - RAMP;

- RANDOM;
- SIN;
- SMOOTH;
- SQRT;
- STEP;
- XIDZ;
- ZIDZ.

Neste contexto, a entrada do simulador é o processo escolhido e as informações relacionadas a ele (as variáveis e os seus valores). Como saída é gerado um conjunto de informações com os detalhes de todas as variáveis utilizadas nas equações, possibilitando diversas análises, inclusive a identificação das melhores e piores configurações e a variável de maior sensibilidade.

Para o cálculo do tempo, leva-se em consideração a variável “média do grau de habilidade do grupo” e uma constante que representa a complexidade para desenvolver cada um dos artefatos. Em se tratando do custo, utiliza-se o resultado obtido com a equação do tempo e a variável “custo por hora” de cada um dos agentes do grupo.

As equações utilizadas para simular o tempo de desenvolvimento do processo são as seguintes:

Tempo para Elaborar os diagramas comportamentais (dias) =  
 SE Média do Grau de Habilidade do Grupo > 5 ENTÃO 6;  
 SE Média do Grau de Habilidade do Grupo > 4 E Média do Grau de Habilidade do Grupo <= 5 ENTÃO 6 \* 1,75;  
 SE Média do Grau de Habilidade do Grupo <= 4 ENTÃO 6 \* 2,25.

Tempo para Elaborar serviços (dias) =  
 SE Média do Grau de Habilidade do Grupo > 5 ENTÃO 6;  
 SE Média do Grau de Habilidade do Grupo > 3 E Média do Grau de Habilidade do Grupo <= 5 ENTÃO 6 \* 1,5;  
 SE Média do Grau de Habilidade do Grupo <= 3 ENTÃO 6 \* 2.

Tempo para Elaborar a interface (dias) =  
 SE Média do Grau de Habilidade do Grupo > 4 ENTÃO 10;  
 SE Média do Grau de Habilidade do Grupo <= 4 ENTÃO 10 \* 1,5.

Tempo para Projetar o diagrama de interação (dias) =  
 SE Média do Grau de Habilidade do Grupo > 4 ENTÃO 5;  
 SE Média do Grau de Habilidade do Grupo <= 4 ENTÃO 5 \* 1,25.

Tempo para Projetar os diagramas estruturais (dias) =  
 SE Média do Grau de Habilidade do Grupo > 5 ENTÃO 13;  
 SE Média do Grau de Habilidade do Grupo > 4 E Média do Grau de Habilidade do Grupo <= 5 ENTÃO 13 \* 1,5;  
 SE Média do Grau de Habilidade do Grupo > 3 E Média do Grau de Habilidade do Grupo <= 4 ENTÃO 13 \* 2;  
 SE Média do Grau de Habilidade do Grupo > 2 E Média do Grau de Habilidade do Grupo <= 3 ENTÃO 13 \* 2,5;  
 SE Média do Grau de Habilidade do Grupo <= 2 ENTÃO 13 \* 3.

Tempo para Projetar testes (dias) =

SE Média do Grau de Habilidade do Grupo > 3 ENTÃO 1;

SE Média do Grau de Habilidade do Grupo <= 3 ENTÃO 1 \* 2.

Tempo para Implementar (dias) =

SE Média do Grau de Habilidade do Grupo > 4,5 ENTÃO 13;

SE Média do Grau de Habilidade do Grupo > 4 E Média do Grau de Habilidade do Grupo <= 4,5 ENTÃO 13 \* 1,25;

SE Média do Grau de Habilidade do Grupo > 3 E Média do Grau de Habilidade do Grupo <= 4 ENTÃO 13 \* 1,75;

SE Média do Grau de Habilidade do Grupo <= 3 ENTÃO 13 \* 2,25.

Tempo total do processo (dias) =

SE Tempo para Elaborar os diagramas comportamentais (dias) >= Tempo para Elaborar serviços (dias) ENTÃO Tempo para Elaborar os diagramas comportamentais (dias);

SE Tempo para Elaborar os diagramas comportamentais (dias) < Tempo para Elaborar serviços (dias) ENTÃO Tempo para Elaborar serviços (dias);

+

Tempo para Elaborar a interface (dias)

+

SE Tempo para Projetar os diagramas estruturais (dias) >= Tempo para Projetar o diagrama de interação (dias) ENTÃO Tempo para Projetar os diagramas estruturais (dias);

SE Tempo para Projetar os diagramas estruturais (dias) < Tempo para Projetar o diagrama de interação (dias) ENTÃO Tempo para Projetar o diagrama de interação (dias);

+

Tempo para Projetar testes (dias)

+

Tempo para Implementar (dias)

As equações utilizadas para simular o custo de desenvolvimento do processo são as seguintes:

Custo total do processo (R\$) =

SE Tempo para Elaborar os diagramas comportamentais (dias) >= Tempo para Elaborar serviços (dias) ENTÃO (Soma do custo por hora do Grupo Diagramas Comportamentais \* Tempo para Elaborar os diagramas comportamentais (dias) \* Horas trabalhadas em um dia) + (Custo por hora do Agente Analista Cx2 \* Tempo para Elaborar serviços (dias) \* Horas trabalhadas em um dia);

SE Tempo para Elaborar os diagramas comportamentais (dias) < Tempo para Elaborar serviços (dias) ENTÃO (Soma do custo por hora do Grupo Serviços \* Tempo para Elaborar os serviços (dias) \* Horas trabalhadas em um dia) + (Custo por hora do Agente Analista Cx1 \* Tempo para Elaborar os diagramas comportamentais (dias) \* Horas trabalhadas em um dia);

+

*Soma do custo por hora do Grupo Interface \* Tempo para Elaborar a interface (dias) \* Horas trabalhadas em um dia*

*+*

*SE Tempo para Projetar os diagramas estruturais (dias) >= Tempo para Projetar o diagrama de interação (dias) ENTÃO (Soma do custo por hora do Grupo Diagramas Estruturais \* Tempo para Projetar os diagramas estruturais (dias) \* Horas trabalhadas em um dia) + (Custo por hora do Agente Analista Sup \* Tempo para Projetar o diagrama de interação (dias) \* Horas trabalhadas em um dia);*

*SE Tempo para Projetar os diagramas estruturais (dias) < Tempo para Projetar o diagrama de interação (dias) ENTÃO (Soma do custo por hora do Grupo Diagrama de Interação \* Tempo para Projetar os diagramas de interação (dias) \* Horas trabalhadas em um dia) + ((Custo por hora do Agente Analista Ctb + Custo por hora do Agente Analista Edu) \* Tempo para Projetar os diagramas estruturais (dias) \* Horas trabalhadas em um dia);*

*+*

*Soma do custo por hora do Grupo Testes \* Tempo para Projetar testes (dias) \* Horas trabalhadas em um dia*

*+*

*Soma do custo por hora do Grupo Implementação \* Tempo para Implementar (dias) \* Horas trabalhadas em um dia*

Com estas equações foi possível executar simulações determinísticas, manipulando os valores atribuídos ao modelo do processo e, ao modificá-los, acompanhar os impactos nos resultados parciais e finais. Simulações estocásticas também puderam ser conduzidas, utilizando os valores reais como limites para a geração de dados aleatórios e a posterior análise das simulações que oferecem os melhores resultados.

Os parâmetros utilizados e os resultados obtidos nas simulações determinísticas e estocásticas são detalhados a seguir, assim como as análises realizadas sobre estas informações, inclusive da sensibilidade.

## **5.2 Resultados obtidos com as simulações determinísticas e estocásticas**

A modelagem do estudo de caso no WebAPSEE com todos os detalhes relacionados às atividades, agentes e artefatos possibilitou ao APSEE-Sim a captura destas informações e a atribuição às variáveis correspondentes, a fim de serem utilizadas nas equações para cada uma das atividades e para o processo. Além da utilização das variáveis, a linguagem Java possibilitou construir as equações com o uso das constantes e dos operadores necessários.

Para a condução de uma simulação determinística com as informações originais, já modeladas no processo, não é necessário nenhum ajuste nas propriedades. O simulador é capaz de utilizar estes valores e, baseando-se nas equações elaboradas, exibir os valores atribuídos às variáveis e calcular o resultado final. A Tabela 5.1 apresenta estes resultados originais.

Tabela 5.4: Resultados da simulação determinística.

SIMULAÇÃO DETERMINÍSTICA – VALORES ORIGINAIS					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	3	180	5,33	6	15.000,00
Elaborar os Serviços	3	175	5	9	
Elaborar a Interface	4	215	5,25	10	17.200,00
Projetar o Diagrama de Interação	3	180	5,33	5	36.320,00
Projetar os Diagramas Estruturais	4	220	4,75	19,5	
Projetar os Testes	3	180	5,33	1	1.440,00
Implementar	13	540	4,54	13	56.160,00
<b>TOTAIS</b>				<b>52,5</b>	<b>126.120,00</b>

Contudo, o gerente pode executar diversas simulações determinísticas, ajustando no arquivo de propriedades os valores desejados para as variáveis. Nas simulações determinísticas os valores das variáveis serão os mesmos para o máximo e mínimo, caso contrário teremos uma simulação estocástica, com a geração de dados aleatórios.

Na Tabela 5.1 é possível observar que não há valor atribuído para o custo das atividades Elaborar os Serviços e Projetar os Diagramas Estruturais. Como estas atividades são desenvolvidas em paralelo e consomem um tempo maior do que as atividades Elaborar os Diagramas Comportamentais e Projetar o Diagrama de Interação, respectivamente, os custos foram atribuídos às atividades que terminam no menor tempo, contudo, contabilizando os custos das duas. Os resultados obtidos na simulação determinística, a partir dos valores originais das variáveis do estudo de caso, serviram de base para a avaliação do desempenho das simulações estocásticas.

Para que as simulações estocásticas pudessem ser desenvolvidas foram atribuídos valores máximos e mínimos para as variáveis utilizadas nas equações. Embora fosse possível, nenhuma das três variáveis teve um valor fixo; para todas elas foram gerados valores aleatórios. Os valores máximos e mínimos foram fixados de acordo com o conhecimento do especialista, que determinou um intervalo possível de ser aplicado na realidade. Embora não seja o caso, nos processos que já possuem dados históricos é possível analisar as informações obtidas das múltiplas execuções para determinar estes valores.

Segundo o especialista os valores máximos e mínimos para cada uma das três variáveis são os seguintes:

- Quantidade de agentes (qag): mínimo igual a dois (qagmin=2) e máximo igual a nove (qagmax=9);
- Soma do custo por hora (sch): mínimo igual a dez (schmin=10) e máximo igual a sessenta (schmax=60);
- Média do grau de habilidade (mha): mínimo igual a dois (mhamin=2) e máximo igual a sete (mhamax=7);

A partir destas informações, o arquivo de propriedades foi “setado” e vinte (20) simulações foram automaticamente executadas. Os resultados foram exportados para uma planilha, conforme a Tabela 5.2, para que pudessem ser analisados.

Tabela 5.2: Resultados da simulação estocástica.

<b>SIMULAÇÃO ESTOCÁSTICA 01</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	8	289,21	3,72	13,5	21.600,00
Elaborar os Serviços	5	216,51	6,21	6	
Elaborar a Interface	5	130,63	6,3	10	17.200,00
Projetar o Diagrama de Interação	2	90,79	5,52	5	36.320,00
Projetar os Diagramas Estruturais	8	325,5	4,89	19,5	
Projetar os Testes	6	221,32	2,94	2	2.880,00
Implementar	7	229,76	4,86	13	56.160,00
<b>TOTAIS</b>				<b>58</b>	<b>134.160,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 02</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	9	312,7	6,69	6	15.000,00
Elaborar os Serviços	9	397,66	3,29	9	
Elaborar a Interface	8	325,13	5,57	10	17.200,00
Projetar o Diagrama de Interação	2	85,06	2,5	6,25	25.380,00
Projetar os Diagramas Estruturais	8	308,32	6,99	13	
Projetar os Testes	4	146,55	5,82	1	1.440,00
Implementar	9	327,88	4,02	16,25	70.200,00
<b>TOTAIS</b>				<b>49,25</b>	<b>129.220,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 03</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	4	188,44	6,3	6	15.000,00
Elaborar os Serviços	8	280,05	3,27	9	
Elaborar a Interface	6	240,84	5	10	17.200,00
Projetar o Diagrama de Interação	5	145,49	2,28	6,25	25.380,00
Projetar os Diagramas Estruturais	4	161,34	6,3	13	
Projetar os Testes	7	208,8	3,69	1	1.440,00
Implementar	3	79,47	6,79	13	56.160,00
<b>TOTAIS</b>				<b>46</b>	<b>115.180,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 04</b>					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	8	300,74	3,34	13,5	23.760,00
Elaborar os Serviços	9	320,21	2,07	12	
Elaborar a Interface	3	142,87	5,64	10	17.200,00
Projetar o Diagrama de Interação	6	240,95	6,3	5	24.880,00
Projetar os Diagramas Estruturais	3	113,97	6,3	13	
Projetar os Testes	6	205,09	3,68	1	1.440,00
Implementar	5	171,63	6,51	13	56.160,00
<b>TOTAIS</b>				<b>50,5</b>	<b>123.440,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 05</b>					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	7	253,42	6,3	6	15.000,00
Elaborar os Serviços	4	147,72	3,22	9	
Elaborar a Interface	7	248,87	2,54	15	25.800,00
Projetar o Diagrama de Interação	6	242,67	6,57	5	24.880,00
Projetar os Diagramas Estruturais	7	223,06	5,49	13	
Projetar os Testes	8	298,04	4,48	1	1.440,00
Implementar	7	276,71	5,28	13	56.160,00
<b>TOTAIS</b>				<b>51</b>	<b>123.280,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 06</b>					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	5	176,47	3,21	13,5	23.760,00
Elaborar os Serviços	3	100,23	2,13	12	
Elaborar a Interface	2	69,3	4,61	10	17.200,00
Projetar o Diagrama de Interação	2	72,35	6,3	5	24.880,00
Projetar os Diagramas Estruturais	5	134,5	5,18	13	
Projetar os Testes	2	103,59	6,3	1	1.440,00
Implementar	6	215,67	6,66	13	56.160,00
<b>TOTAIS</b>				<b>50,5</b>	<b>123.440,00</b>



<b>SIMULAÇÃO ESTOCÁSTICA 07</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	3	92,23	5,26	6	10.800,00
Elaborar os Serviços	4	148,86	5,92	6	
Elaborar a Interface	6	202,25	2,03	15	25.800,00
Projetar o Diagrama de Interação	6	216,63	5,11	5	47.760,00
Projetar os Diagramas Estruturais	7	209,66	3,97	26	
Projetar os Testes	9	248,15	5,55	1	1.440,00
Implementar	6	209,27	4,12	16,25	70.200,00
<b>TOTAIS</b>				<b>64,25</b>	<b>156.000,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 08</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	3	59,05	6,28	6	19.200,00
Elaborar os Serviços	4	140,15	2,94	12	
Elaborar a Interface	2	74,78	6,19	10	17.200,00
Projetar o Diagrama de Interação	2	83,43	6,07	5	36.320,00
Projetar os Diagramas Estruturais	7	251,99	4,3	19,5	
Projetar os Testes	6	231,29	6,49	1	1.440,00
Implementar	7	250,38	4,24	16,25	70.200,00
<b>TOTAIS</b>				<b>58,75</b>	<b>144.360,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 09</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	8	370,25	4,26	10,5	18.360,00
Elaborar os Serviços	2	93	3,11	9	
Elaborar a Interface	4	103,05	5,92	10	17.200,00
Projetar o Diagrama de Interação	8	229,41	4,91	5	47.760,00
Projetar os Diagramas Estruturais	3	106,45	3,43	26	
Projetar os Testes	7	278,39	2,99	2	2.880,00
Implementar	8	264,11	6,3	13	56.160,00
<b>TOTAIS</b>				<b>61,5</b>	<b>142.360,00</b>

SIMULAÇÃO ESTOCÁSTICA 10					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	2	45,07	4,95	10,5	21.000,00
Elaborar os Serviços	3	79,53	2,7	12	
Elaborar a Interface	9	312,31	5,69	10	17.200,00
Projetar o Diagrama de Interação	6	233,29	6,3	5	24.880,00
Projetar os Diagramas Estruturais	6	223,42	5,69	13	
Projetar os Testes	4	173,57	3,42	1	1.440,00
Implementar	9	345,12	6,14	13	56.160,00
<b>TOTAIS</b>				<b>49</b>	<b>120.680,00</b>

SIMULAÇÃO ESTOCÁSTICA 11					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	8	230,19	6,3	6	15.000,00
Elaborar os Serviços	5	209,45	3,39	9	
Elaborar a Interface	3	129,45	4,21	10	17.200,00
Projetar o Diagrama de Interação	4	167,76	3,96	6,25	48.260,00
Projetar os Diagramas Estruturais	3	121,07	3,43	26	
Projetar os Testes	3	163,02	4,23	1	1.440,00
Implementar	3	108,39	4,24	16,25	70.200,00
<b>TOTAIS</b>				<b>62,25</b>	<b>152.100,00</b>

SIMULAÇÃO ESTOCÁSTICA 12					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	9	369,57	6,59	6	15.000,00
Elaborar os Serviços	2	36,46	4,37	9	
Elaborar a Interface	2	89,11	3,36	15	25.800,00
Projetar o Diagrama de Interação	6	218,85	6,3	5	24.880,00
Projetar os Diagramas Estruturais	7	280,77	6,3	13	
Projetar os Testes	3	83,24	6,59	1	1.440,00
Implementar	8	247,26	3,9	22,75	98.280,00
<b>TOTAIS</b>				<b>60,75</b>	<b>165.400,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 13</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	7	299,14	6,3	6	15.000,00
Elaborar os Serviços	6	234,19	4,8	9	
Elaborar a Interface	3	130,09	6,3	10	17.200,00
Projetar o Diagrama de Interação	5	217,32	3,9	6,25	48.260,00
Projetar os Diagramas Estruturais	3	103,36	3,18	26	
Projetar os Testes	3	105,59	6,3	1	1.440,00
Implementar	9	363,5	5,04	13	56.160,00
<b>TOTAIS</b>				<b>59</b>	<b>138.060,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 14</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	4	128,46	6,76	6	15.000,00
Elaborar os Serviços	7	284,71	3,28	9	
Elaborar a Interface	5	210,16	5,95	10	17.200,00
Projetar o Diagrama de Interação	9	338,12	6,3	5	36.320,00
Projetar os Diagramas Estruturais	8	301,48	4,05	19,5	
Projetar os Testes	6	239,95	6,3	1	1.440,00
Implementar	7	197,24	3,97	22,75	98.280,00
<b>TOTAIS</b>				<b>62,25</b>	<b>168.240,00</b>

<b>SIMULAÇÃO ESTOCÁSTICA 15</b>					
<b>Atividade</b>	<b>Qtd. de Agentes</b>	<b>Soma Custo HR</b>	<b>Média Habil.</b>	<b>Tempo (Dias)</b>	<b>Custo (R\$)</b>
Elaborar os Diagramas Comportamentais	8	274,18	5,92	6	10.800,00
Elaborar os Serviços	3	114,43	6,68	6	
Elaborar a Interface	6	192,19	3,71	15	25.800,00
Projetar o Diagrama de Interação	2	76,82	6,76	5	24.880,00
Projetar os Diagramas Estruturais	4	112,58	6,26	13	
Projetar os Testes	8	249,27	5,67	1	1.440,00
Implementar	5	147,43	6,28	13	56.160,00
<b>TOTAIS</b>				<b>48</b>	<b>119.080,00</b>

SIMULAÇÃO ESTOCÁSTICA 16					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	7	281,84	6,98	6	15.000,00
Elaborar os Serviços	6	244,01	4,32	9	
Elaborar a Interface	9	284,41	6,22	10	17.200,00
Projetar o Diagrama de Interação	7	283,73	2,5	6,25	25.380,00
Projetar os Diagramas Estruturais	7	279,98	6,3	13	
Projetar os Testes	9	283,87	5,12	1	1.440,00
Implementar	8	305,6	3,95	22,75	98.280,00
<b>TOTAIS</b>				<b>55,75</b>	<b>157.300,00</b>

SIMULAÇÃO ESTOCÁSTICA 17					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	4	169,34	5,54	6	15.000,00
Elaborar os Serviços	7	230,35	3,12	9	
Elaborar a Interface	5	175,46	3,92	15	25.800,00
Projetar o Diagrama de Interação	4	168,59	2,39	6,25	59.700,00
Projetar os Diagramas Estruturais	8	263,32	2,07	32,5	
Projetar os Testes	6	171,16	3,08	1	1.440,00
Implementar	3	93,54	5,87	13	56.160,00
<b>TOTAIS</b>				<b>70,5</b>	<b>158.100,00</b>

SIMULAÇÃO ESTOCÁSTICA 18					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	5	176,46	3,74	13,5	21.600,00
Elaborar os Serviços	2	79,32	6,94	6	
Elaborar a Interface	8	231,36	6,81	10	17.200,00
Projetar o Diagrama de Interação	8	298,47	6,3	5	24.880,00
Projetar os Diagramas Estruturais	8	288,68	6,3	13	
Projetar os Testes	9	269	4,1	1	1.440,00
Implementar	7	206,29	6,3	13	56.160,00
<b>TOTAIS</b>				<b>50,5</b>	<b>121.280,00</b>

SIMULAÇÃO ESTOCÁSTICA 19					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	3	97,34	6,3	6	10.800,00
Elaborar os Serviços	4	147,23	6,3	6	
Elaborar a Interface	3	83,44	6,01	10	17.200,00
Projetar o Diagrama de Interação	7	209,91	6,3	5	24.880,00
Projetar os Diagramas Estruturais	6	195,32	6,3	13	
Projetar os Testes	8	246,23	4,08	1	1.440,00
Implementar	8	313,12	2,79	29,25	126.360,00
<b>TOTAIS</b>				<b>59,25</b>	<b>180.680,00</b>

SIMULAÇÃO ESTOCÁSTICA 20					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	3	87,46	3,86	13,5	22.680,00
Elaborar os Serviços	8	303,12	3,2	9	
Elaborar a Interface	8	273,11	5,01	10	17.200,00
Projetar o Diagrama de Interação	6	169,64	5,54	5	47.760,00
Projetar os Diagramas Estruturais	8	223,32	3,54	26	
Projetar os Testes	9	276,87	4,91	1	1.440,00
Implementar	8	282,62	3,14	22,75	98.280,00
<b>TOTAIS</b>				<b>73,25</b>	<b>187.360,00</b>

Na Tabela 5.2 foram destacados os resultados que indicam uma melhoria no tempo e/ou custo em relação aos valores originalmente modelados, conforme Tabela 5.1: Tempo original igual a cinquenta e dois dias e meio (Tempo=52,5) e custo original igual a cento e vinte e seis mil e cento e vinte reais (Custo=R\$126.120,00).

Neste contexto, também foi possível verificar qual o melhor e o pior caso. De acordo com as vinte (20) simulações realizadas, o pior tempo e o custo mais elevado é verificado na vigésima (20ª) simulação, com os valores de setenta e três dias e duas horas (73,25) e cento e oitenta e sete reais, trezentos e sessenta centavos (R\$187.360,00). Em contrapartida, o menor tempo e o custo mais baixo é verificado na terceira (3ª) simulação, com quarenta e seis (46) dias e cento e quinze mil e cento e oitenta reais (R\$115.180,00).

Estes resultados podem ser melhor visualizados nos gráficos abaixo. A Figura 5.2 apresenta os valores obtidos para o tempo e a Figura 5.3, para o custo. Em cada um dos gráficos foi traçada uma linha que representa o tempo e o custo que o estudo de caso consumiu na realidade e foram comprovados pela simulação determinística.

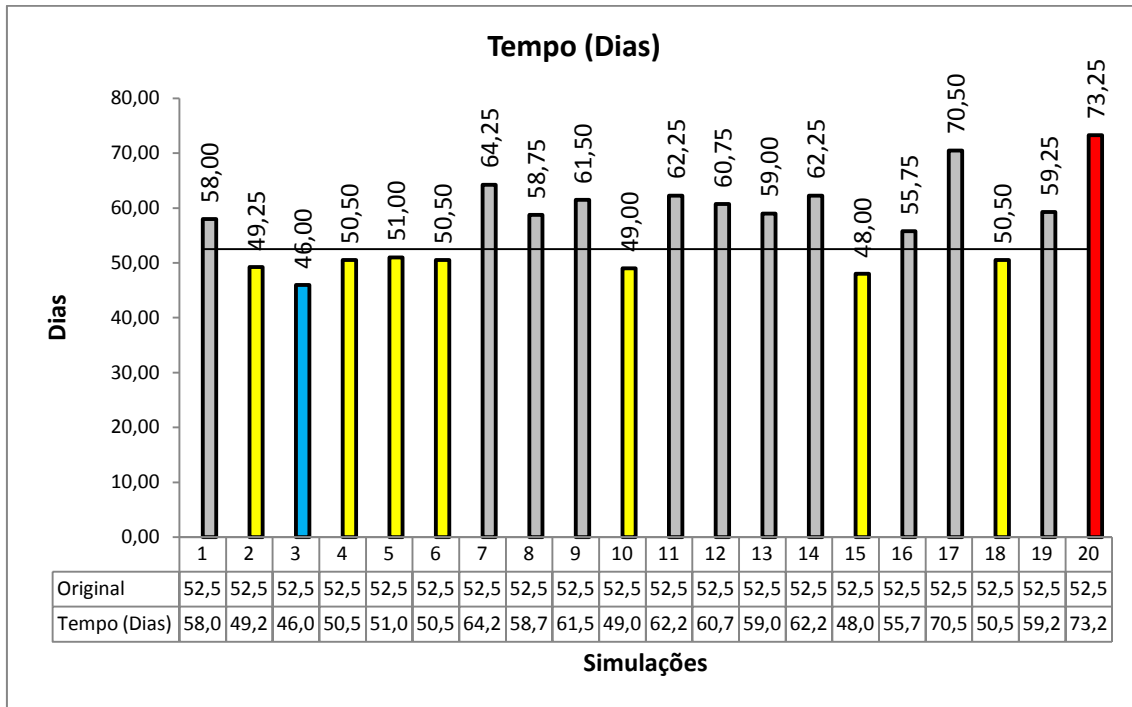


Figura 5.2: Resultados obtidos na simulação do tempo em dias.

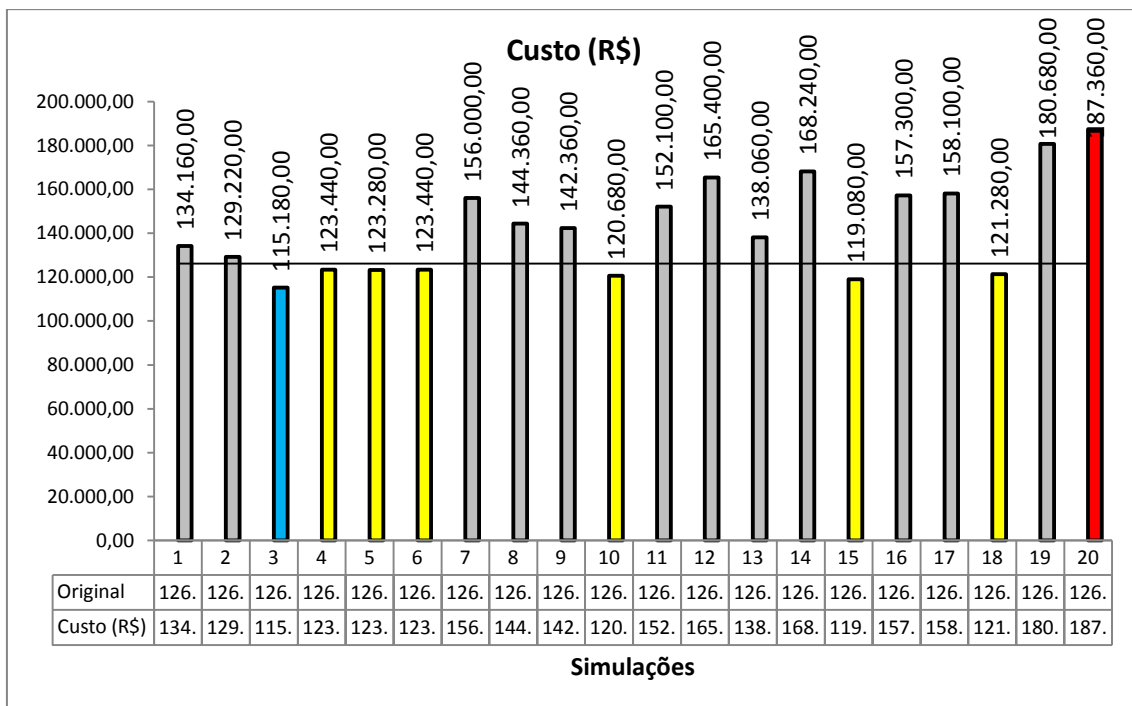


Figura 5.3: Resultados obtidos na simulação do custo em reais.

A partir dos resultados obtidos foi possível identificar oito simulações que indicam uma redução do tempo e sete simulações que indicam uma redução do custo em relação ao estudo de caso original. Cada uma destas simulações leva em consideração valores específicos para as variáveis das atividades. Os detalhes de cada uma destas variáveis estão representados no Tabela 5.2 e, com a posse destas informações, o gerente pode

escolher a simulação que melhor se adapta à realidade do projeto e, caso desejar, alterar a modelagem do processo antes da execução do mesmo.

Apesar dos valores máximos e mínimos atribuídos para cada uma das variáveis terem sido determinados de acordo com a realidade do projeto, é necessário que o gerente analise e indique qual simulação deverá ser escolhida, afinal, nem sempre a simulação que oferece os melhores resultados pode ser modelada e simulada. Por exemplo, na terceira simulação, que indica o consumo do menor tempo e custo, são exigidos quatro (4) agentes, que consomem no máximo cento e oitenta e oito reais e quarenta e quatro centavos (R\$188,44) por hora e possuem uma média de seis vírgula três (6,3) no grau de habilidade para o desenvolvimento da atividade Elaborar os Diagramas Comportamentais. Caso esta configuração não seja possível devido a fatores externos, como a indisponibilidade da quantidade sugerida de agentes, ou que consumam valores inferior ou iguais aos indicados, entre outros, pode-se utilizar os valores sugeridos para as variáveis da próxima simulação mais otimizada.

Utilizando a técnica Monte Carlo na simulação é possível aprimorar os valores máximos e mínimos (intervalos). Desta forma, se nenhuma das configurações resultantes puder ser aplicada, a alternativa será conduzir simulações determinísticas com valores aproximados aos disponíveis ou simulações estocásticas com valores otimizados para os máximos e mínimos, além de uma quantidade maior de iterações, até que seja possível encontrar alguma situação que otimize o estudo de caso e possa ser concretizada.

Os resultados obtidos para os custos se assemelham aos tempos, devido ao relacionamento existente entre estas variáveis na elaboração das equações. Contudo, a exceção ocorre na segunda simulação, que reflete uma redução de tempo com um custo mais elevado em relação aos parâmetros de base.

Com os resultados da simulação o gerente poderá aplicar os parâmetros da simulação otimista, qualificando o processo. Caso a aplicação destes parâmetros tornar-se inviável, será possível escolher outro cenário otimista para ser utilizado. Se nenhuma simulação otimista puder ser aplicada no processo real, a execução de novas simulações poderá ser realizada, até que seja possível encontrar um cenário que otimize o processo. De acordo com a análise do gerente, quatro (4) simulações resultantes poderiam ser concretizadas, reduzindo os tempos e custos originalmente consumidos.

### **5.3 Análise da sensibilidade a partir dos resultados obtidos**

Com os recursos oferecidos pelo simulador, foi possível conduzir simulações que permitiram realizar a análise da sensibilidade a fim de identificar a variável de maior impacto no resultado final.

A primeira variável analisada foi a Quantidade de agentes (QAG). Esta teve uma variação nos seus valores, dentro do intervalo permitido e estipulado pelo gerente, enquanto as duas outras permaneceram inalteradas: Soma do custo por hora (SCH) e Média do grau de habilidade (MHA).

Neste contexto, a simulação resultou em vinte (20) iterações utilizando os seguintes parâmetros:

- qagmin=2
- qagmax=9
- sch=35

- mha=4,5

A Tabela 5.3 mostra o resultado de uma destas simulações, na qual a Quantidade de agentes (QAG) varia de forma aleatória, a Soma do custo por hora (SCH) representa a multiplicação da Quantidade de agentes (QAG) por trinta e cinco reais (R\$35) e a Média do grau de habilidades (MHA) permanece inalterado.

Tabela 5.3: Resultado da análise da sensibilidade em função da quantidade de agentes (QAG)

SIMULAÇÃO ESTOCÁSTICA 01					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	8	280,00	4,50	10,50	18.360,00
Elaborar os Serviços	5	175,00	4,50	9,00	
Elaborar a Interface	5	175,00	4,50	10,00	17.200,00
Projetar o Diagrama de Interação	2	70,00	4,50	5,00	36.320,00
Projetar os Diagramas Estruturais	8	280,00	4,50	19,50	
Projetar os Testes	6	210,00	4,50	1,00	1.440,00
Implementar	7	245,00	4,50	16,25	70.200,00
<b>TOTAIS</b>				<b>57,25</b>	<b>143.520,00</b>

Outra variável analisada foi a Soma do custo por hora (SCH). Esta teve uma variação nos seus valores, dentro do intervalo permitido e estipulado pelo gerente, enquanto as duas outras permaneceram inalteradas: Quantidade de agentes (QAG) e Média do grau de habilidade (MHA).

Neste contexto, a simulação resultou em vinte (20) iterações utilizando os seguintes parâmetros:

- qag=5
- schmin=10
- schmax=60
- mha=4,5

A Tabela 5.4 mostra o resultado de uma destas simulações, na qual a Soma do custo por hora (SCH) varia de forma aleatória, a Quantidade de agentes (QAG) e a Média do grau de habilidades (MHA) permanece inalterado.



Tabela 5.4: Resultado da análise da sensibilidade em função da soma do custo por hora (SCH)

SIMULAÇÃO ESTOCÁSTICA 01					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	5	289,21	4,50	10,50	18.360,00
Elaborar os Serviços	5	216,51	4,50	9,00	
Elaborar a Interface	5	130,63	4,50	10,00	17.200,00
Projetar o Diagrama de Interação	5	90,79	4,50	5,00	36.320,00
Projetar os Diagramas Estruturais	5	325,50	4,50	19,50	
Projetar os Testes	5	221,32	4,50	1,00	1.440,00
Implementar	5	229,76	4,50	16,25	70.200,00
<b>TOTAIS</b>				<b>57,25</b>	<b>143.520,00</b>

A terceira variável analisada foi a Média do grau de habilidade (MHA). Esta teve uma variação nos seus valores, dentro do intervalo permitido e estipulado pelo gerente, enquanto as duas outras permaneceram inalteradas: Quantidade de agentes (QAG) e Soma do custo por hora (SCH).

Neste contexto, a simulação resultou em vinte (20) iterações utilizando os seguintes parâmetros:

- qag=5
- sch=35
- mhamin=2
- mhamin=7

A Tabela 5.5 mostra o resultado de uma destas simulações, na qual a Média do grau de habilidades (MHA) varia de forma aleatória, a Quantidade de agentes (QAG) e a Soma do custo por hora (SCH) permanece inalterado.

Tabela 5.5: Resultado da análise da sensibilidade em função da média do grau de habilidades (MHA)

SIMULAÇÃO ESTOCÁSTICA 01					
Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Tempo (Dias)	Custo (R\$)
Elaborar os Diagramas Comportamentais	5	175,00	3,72	13,50	21.600,00
Elaborar os Serviços	5	175,00	6,21	6,00	
Elaborar a Interface	5	175,00	6,30	10,00	17.200,00
Projetar o Diagrama de Interação	5	175,00	5,52	5,00	36.320,00
Projetar os Diagramas Estruturais	5	175,00	4,89	19,50	
Projetar os Testes	5	175,00	2,94	2,00	2.880,00
Implementar	5	175,00	4,86	13,00	56.160,00
<b>TOTAIS</b>				<b>58,00</b>	<b>134.160,00</b>

Os resultados obtidos com as vinte (20) simulações, para cada uma das variáveis enquanto as outras permaneciam inalteradas, foram organizados nos gráficos da Figura 5.4 para o tempo e Figura 5.5 para o custo.

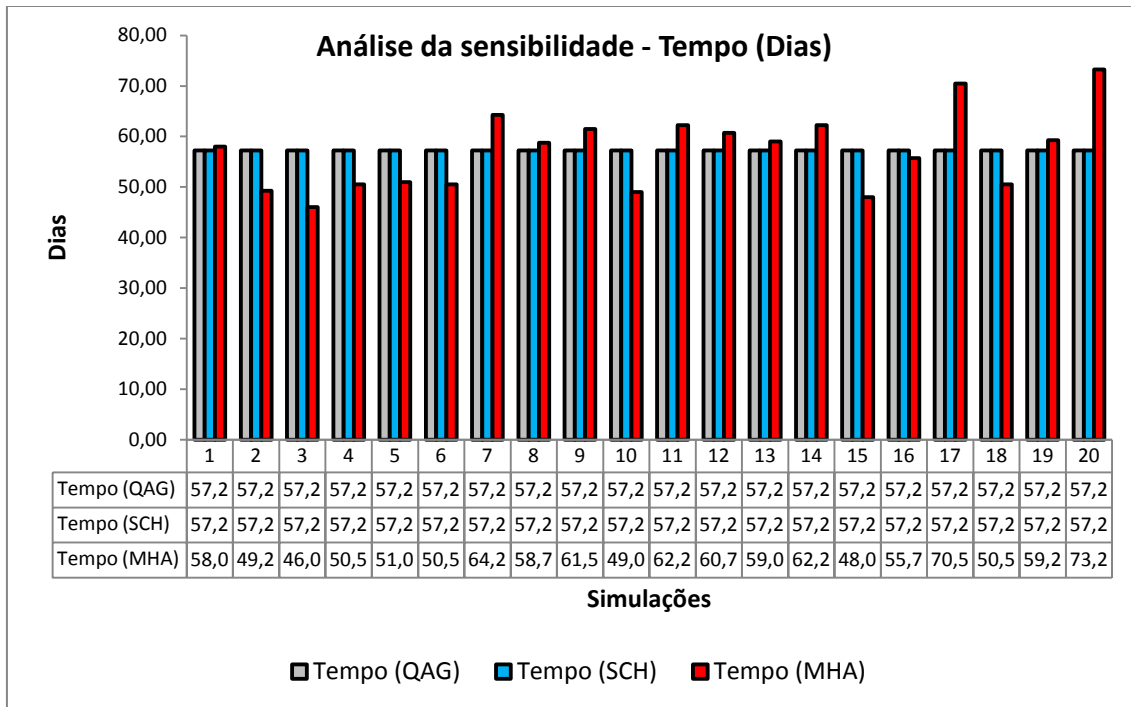


Figura 5.4: Análise da sensibilidade do tempo em dias

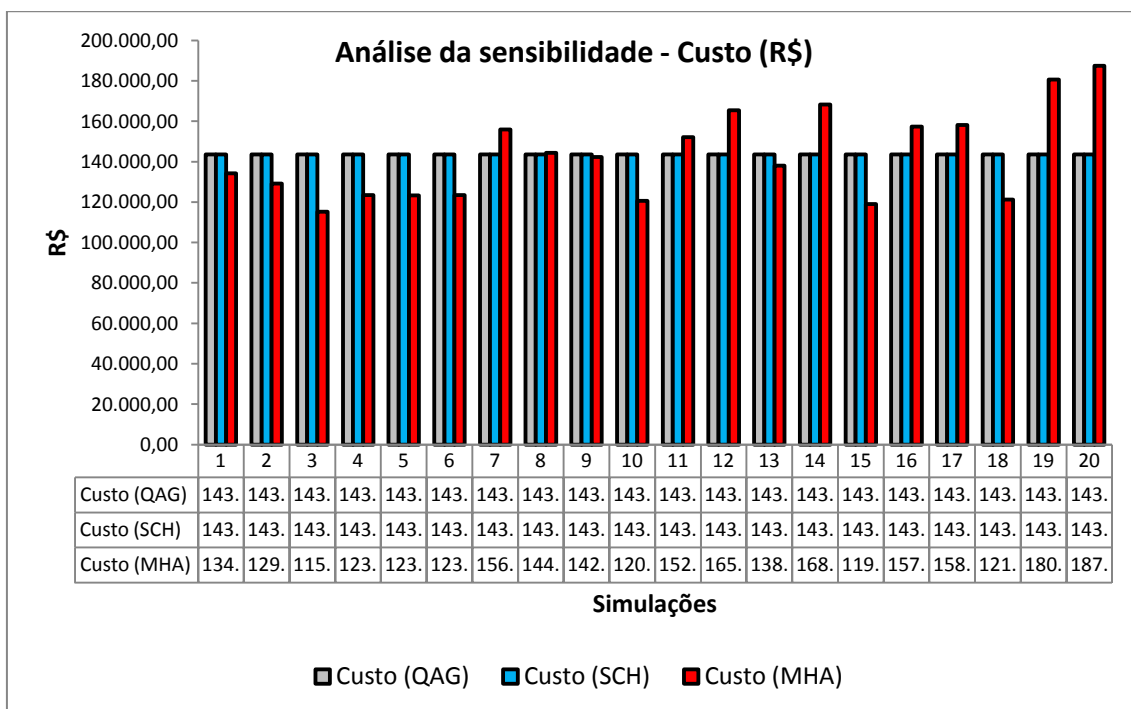


Figura 5.5: Análise da sensibilidade do custo em reais

Nas simulações que tiveram valores aleatórios para a variável Quantidade de agentes (QAG) e as demais variáveis (SCH e MHA) permaneceram inalteradas, os resultados das equações para o tempo e custo permaneceram inalterados. Ou seja, a quantidade de agentes não influencia diretamente no resultado final.

A mesma conclusão pode ser obtida com as simulações que tiveram valores aleatórios para a variável Soma do custo por hora (SCH) e as demais variáveis (QAG e MHA) permaneceram inalteradas, os resultados das equações para o tempo e custo permaneceram inalterados. Ou seja, a soma do custo por hora não influencia diretamente no resultado final.

Nas simulações que tiveram valores aleatórios para a variável Média do grau de habilidades (MHA) e as demais variáveis (QAG e SCH) permaneceram inalteradas, os resultados das equações para o tempo e custo tiveram significativas alterações. Ou seja, a média do grau de habilidade é a variável de maior sensibilidade e influencia diretamente no resultado final.

Desta forma, mesmo que o gerente aloque um número reduzido de agentes, mas escolha aqueles com uma média alta do grau de habilidades, sem extrapolar a soma do custo por hora, o resultado final será um tempo e custo próximos aos ideais.

#### **5.4 Avaliação dos recursos oferecidos pelo simulador**

Com o objetivo de avaliar os recursos implementados no simulador proposto, foi realizada uma verificação considerando características da modelagem, simulação e resultados. A tabela que sintetiza as informações obtidas segue o padrão utilizado no capítulo 3, que considera as informações como presentes (✓), ausentes (✗) ou não foi possível avaliar (-).

O estudo de caso simulado pelo APSEE-Sim foi modelado utilizando os recursos disponíveis no ambiente WebAPSEE, conforme a Figura 5.1, alimentando a base de dados que será utilizada como entrada para as operações do simulador.

O simulador não possui recursos para a elaboração dinâmica de equações para as atividades e processos, porém, esta deficiência pode ser contornada inserindo-se as equações no próprio código-fonte, na classe *Simulate.java*, conforme a Figura 5.6. Mais detalhes sobre esta classe podem ser consultados no Anexo B.

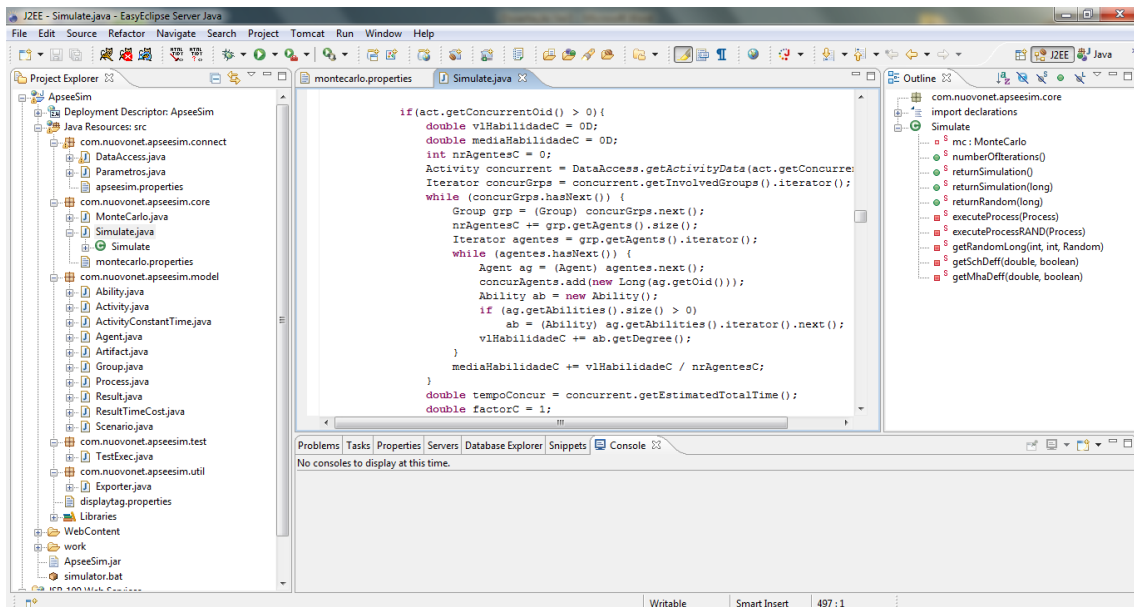


Figura 5.6: Equações elaboradas no *Easy Eclipse*

O APSEE-Sim pode ser acessado a partir de um navegador, conforme Figura 5.7, e disponibiliza recursos para escolher o processo a ser simulado, visualizar os resultados de uma simulação determinística através do menu *Simulation Details*, ou estocástica através da opção Monte Carlo.

Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
Estudo de caso.Elaborar os Diagramas Comportamentais	13	R\$ 150,40	1,27	0	2	13,50	R\$ 16.242,94
Estudo de caso.Elaborar os Serviços	5	R\$ 180,07	3,76	0	2	9,00	R\$ 12.964,63
Estudo de caso.Elaborar a Interface	15	R\$ 207,61	1,27	0	2	15,00	R\$ 24.912,67
Estudo de caso.Projetar o Diagrama de Interação	5	R\$ 164,67	2,32	0	1	0,00	R\$ 2.442,47
Estudo de caso.Projetar os Diagramas Estruturais	18	R\$ 241,00	1,02	0	4	39,00	R\$ 75.190,82
Estudo de caso.Projetar os Testes	18	R\$ 153,67	0,56	0	1	2,00	R\$ 2.458,71
Estudo de caso.Implementar	14	R\$ 709,45	4,19	0	3	16,25	R\$ 92.228,80
<b>T O T A L</b>						<b>94,75</b>	<b>R\$ 226.441,22</b>

Atividade	Qtd. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
Estudo de caso.Elaborar os Diagramas Comportamentais	14	R\$ 164,48	1,08	0	2	13,50	R\$ 17.764,09
Estudo de caso.Elaborar os Serviços	10	R\$ 176,40	1,10	0	2	12,00	R\$ 16.934,31
Estudo de caso.Elaborar a Interface	5	R\$ 231,95	3,58	0	2	15,00	R\$ 27.834,07
Estudo de caso.Projetar o Diagrama de Interação	18	R\$ 167,93	0,56	0	1	0,00	R\$ 3.251,62
Estudo de caso.Projetar os Diagramas Estruturais	4	R\$ 200,69	4,46	0	4	19,50	R\$ 31.307,44
Estudo de caso.Projetar os Testes	8	R\$ 172,61	1,42	0	1	2,00	R\$ 2.761,79
Estudo de caso.Implementar	14	R\$ 728,39	4,90	0	3	13,00	R\$ 75.752,46
<b>T O T A L</b>						<b>75,00</b>	<b>R\$ 175.605,78</b>

Figura 5.7: *Mozilla Firefox* com os recursos do APSEE-Sim

Além de visualizar as informações, o APSEE-Sim permite exportar as informações geradas para uma planilha eletrônica, conforme a Figura 5.8, que oferece vastos recursos para que estes resultados sejam analisados e contextualizados na forma de relatórios ou gráficos, como relatado anteriormente.

Atividade	Qt. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
<b>1ª Simulation Randomic Totals</b>							
Atividade	Qt. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
Estudo de caso.Elaborar os Diagramas Comportamentais	11	100,01	1,05	0	2	13,5	10.800,57
Estudo de caso.Elaborar os Serviços	13	125,24	0,99	0	2	12	12.022,99
Estudo de caso.Elaborar a Interface	6	125,23	4,53	0	2	10	10.018,44
Estudo de caso.Projetar o Diagrama de Interação	13	129,01	1,58	0	1	0	1.147,82
Estudo de caso.Projetar os Diagramas Estruturais	12	195,53	0,8	0	4	39	61.005,96
Estudo de caso.Projetar os Testes	12	72,18	1,33	0	1	2	1.154,93
Estudo de caso.Implementar	9	457,01	8,91	0	3	13	47.529,19
<b>1ª Simulation Randomic Totals</b>						<b>89,5</b>	<b>143.679,90</b>
<b>2ª Simulation Randomic Totals</b>							
Atividade	Qt. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
Estudo de caso.Elaborar os Diagramas Comportamentais	7	107,05	2,29	0	2	13,5	11.561,06
Estudo de caso.Elaborar os Serviços	8	121,94	1,8	0	2	12	11.706,04
Estudo de caso.Elaborar a Interface	13	110,48	1,5	0	2	15	13.257,42
Estudo de caso.Projetar o Diagrama de Interação	9	114,63	1,73	0	1	0	2.071,14
Estudo de caso.Projetar os Diagramas Estruturais	13	185,28	1,18	0	4	39	57.808,88
Estudo de caso.Projetar os Testes	6	144,87	2,19	0	1	2	2.317,89
Estudo de caso.Implementar	12	470,33	5,9	0	3	13	48.914,16
<b>2ª Simulation Randomic Totals</b>						<b>94,5</b>	<b>147.636,58</b>
<b>3ª Simulation Randomic Totals</b>							
Atividade	Qt. de Agentes	Soma Custo HR	Média Habil.	Artefatos Entrada	Artefatos Saída	Tempo (Dias)	Custo (R\$)
Estudo de caso.Elaborar os Diagramas Comportamentais	7	141,05	1,44	0	2	13,5	15.233,93
Estudo de caso.Elaborar os Serviços	5	102,21	3,14	0	2	0	1.725,06
Estudo de caso.Elaborar a Interface	11	159,61	2	0	2	15	19.153,76
Estudo de caso.Projetar o Diagrama de Interação	13	127,76	0,74	0	1	0	1.050,93
Estudo de caso.Projetar os Diagramas Estruturais	13	153,49	1,89	0	4	39	47.890,02
Estudo de caso.Projetar os Testes	10	84,41	0,94	0	1	2	1.350,53

Figura 5.8: *MS-Excel* com os resultados exportados do APSEE-Sim

Com a modelagem do estudo de caso, incluindo as equações para as atividades e para o processo, foi possível realizar diversas simulações, apurar os resultados obtidos e, neste contexto, vários fatores foram avaliados e contextualizados na Tabela 5.6, que foi idealizada no capítulo 3 deste trabalho.

Tabela 5.6: Características avaliadas no APSEE-Sim.

CARACTERÍSTICAS AVALIADAS	APSEE-Sim	CARACTERÍSTICAS AVALIADAS	APSEE-Sim	CARACTERÍSTICAS AVALIADAS	APSEE-Sim
<b>MODELAGEM</b>		<b>SIMULAÇÃO</b>		<b>RESULTADOS</b>	
Permite criar o processo graficamente	✓	Simula um domínio específico	✓	Permite armazenar os resultados obtidos	✓
Permite a elaboração de equações	✓	Possibilita estratégias de simulação discreta	✓	Possibilita a personalização de relatórios	✓
Possibilita a visualização da hierarquia de execução	✓	Possibilita estratégias de simulação contínua	✗	Dispõe de recursos para que os dados sejam analisados	✓
Dispõe de exemplos de processos	✓	Permite a simulação de modelos determinísticos	✓	Identifica áreas que podem ser melhoradas	✓
Permite a reutilização de processos	✓	Permite a simulação de modelos estocásticos	✓		
Permite importar modelos de outros programas	✗	Permite simular diferentes cenários	✓		
		Dispõe de recursos de animação	✗		

Na tabela resultante é possível verificar que devido à característica do simulador atuar junto ao WebAPSEE, ele permite que o usuário utilize os recursos deste PSEE para criar o processo graficamente, visualizar a hierarquia de execução e reutilizar processos.

No APSEE-Sim a inserção das equações nas atividades é realizada no próprio código-fonte. Conforme ilustra a Figura 5.6, esta inserção foi realizada no ambiente de desenvolvimento *Easy Eclipse*.

O estudo de caso apresentado neste trabalho é um exemplo de processo e pode ser reutilizado. Todo processo modelado no WebAPSEE pode ser simulado a partir de equações inseridas no código-fonte e os resultados obtidos armazenados e utilizados para diversos propósitos: melhoria do processo, condução de novas simulações, etc.

Como o WebAPSEE não permite exportar modelos de outros softwares, esta característica também não está disponível no simulador. O processo precisa ser

modelado em detalhes no PSEE e em seguida simulado. Esta é a situação da grande maioria dos simuladores para todos os tipos de processos.

O APSEE-Sim foi desenvolvido como um simulador de processos de software, incorporado a um PSEE e, de acordo com esta característica, está restrito a este domínio. Desta forma, possibilita uma consistência elevada acerca do domínio proposto.

Como o modelo do simulador foi projetado para atuar entre a modelagem e a execução, na simulação o tempo é estático e, desta forma, tem-se uma simulação discreta. A execução da simulação durante a execução do processo, em tempo real, permitiria uma estratégia de simulação contínua e exigiria adaptações no modelo.

O simulador foi concebido para permitir a condução de simulações determinísticas, com valores pontualmente definidos e ainda, simulações estocásticas, através da geração de valores aleatórios, dentro de um intervalo definido e na quantidade desejada de iterações. Com esta proposta também é possível realizar a simulação de diversos cenários (tempo, custo, etc.).

Como a implementação do simulador objetivou provar um conceito (modelo), não foram desenvolvidos avançados recursos gráficos e, desta forma, o APSEE-Sim não dispõem de recursos de animação.

Os resultados obtidos com as simulações podem ser visualizados no próprio *browser*, conforme ilustrado na Figura 5.7, ou podem ser exportados para uma planilha eletrônica, de acordo com a Figura 5.8. Os dados exportados exibem as atividades do processo, as variáveis e os valores atribuídos. Com os recursos oferecidos pelas planilhas eletrônicas, o gerente poderá criar relatórios personalizados, utilizar fórmulas e funções para auxiliar na análise destes resultados, tomar decisões nos casos críticos, identificar as áreas que podem ser melhoradas e armazenar os resultados obtidos.

## 6 CONCLUSÃO

Este trabalho apresentou um modelo de simulador que permite extrair informações dos processos modelados no PSEE, atribuir equações para as atividades na condução de simulações determinísticas e estocásticas, gerar dados para a condução de diversas análises, inclusive da sensibilidade, identificando as melhores configurações para aprimorar a modelagem do processo.

A seguir, são apresentadas as principais contribuições do trabalho, suas limitações e trabalhos futuros.

### 6.1 Contribuições

Os resultados obtidos com o desenvolvimento da revisão sistemática, acerca dos trabalhos publicados sobre simulação nos últimos anos, permitiram estruturar e mapear os conhecimentos sobre o domínio da simulação, sendo útil para abreviar futuros trabalhos e estabelecer relações entre eles. Desta forma, representa uma das contribuições deste trabalho e, principalmente, serviu como base para o desenvolvimento dos conceitos abordados no modelo do simulador.

A avaliação comparativa dos simuladores identificados com a revisão sistemática também pode ser apontada como contribuição. Esta avaliação permitiu verificar o nível de apoio tecnológico oferecido pelos softwares e, desta forma, apontar as ferramentas mais indicadas para resolver determinado problema, os pontos fortes e fracos de cada uma delas e os recursos a serem contemplados na proposta do modelo apresentado neste estudo.

Neste contexto, a maior contribuição deste trabalho é o modelo sugerido e a ideia do quê a simulação pode proporcionar, como é possível projetar um simulador extraíndo informações automáticas a partir do PSEE, para a condução de simulações determinísticas e estocásticas, utilizando a técnica Monte Carlo. Apesar de ser genérico e descrito em alto nível de abstração, é capaz de trazer algo novo para o desenvolvimento de software, muito mais do que a implementação em si, que se fez necessária para provar o conceito proposto e através dos resultados demonstrou o potencial da solução para comercialização.

Como o modelo de simulador está integrado a um PSEE, não é necessário modelar o processo. As informações são extraídas automaticamente (variáveis) e, ao inserir as equações, permite executar uma simulação determinística com os valores já atribuídos ao processo, sem a necessidade de nenhum ajuste. Alterando-se os valores máximos e mínimos das variáveis é possível conduzir simulações determinísticas e estocásticas, gerando dados para serem manipulados em planilhas eletrônicas, que oferecem os recursos necessários para a condução de diversas análises, inclusive da sensibilidade. Nos simuladores de propósito geral é necessário modelar o processo na ferramenta,



ocorrendo perda de informações detalhadas que estão modeladas no PSEE. Com os simuladores incorporados ao PSEE, como no modelo proposto, esta perda não ocorre.

Com os resultados das simulações foi possível identificar, no estudo de caso, as configurações que indicam melhorias no processo em termos de custo e tempo e é apontada a variável de maior impacto no resultado final, permitindo desta forma, otimizações e melhorias na modelagem. A condução da avaliação comparativa e a experimentação demonstrou que um simulador incorporado a um PSEE é capaz de oferecer informações mais precisas sobre o processo, quando comparado a um simulador de propósito geral.

## 6.2 Limitações

Na implementação, identifica-se, como limitação deste trabalho, a falta de uma solução para facilitar a elaboração e edição de equações através da interface *web*. Existem simuladores que permitem que as equações sejam registradas como parte do processo de modelagem, sem exigir a alteração do código-fonte. Contudo, esta característica não iria interferir nos resultados obtidos, apenas melhorar a usabilidade da ferramenta.

Para que o modelo de simulador proposto possa ser implementado como complemento de um PSEEs, o mesmo deve possibilitar a extração das informações dos processos a partir de sua base de dados.

O suporte ferramental de apoio ao gerente na análise dos dados gerados com as simulações, para que identifique as melhores configurações e aprimore o processo, é outro aspecto da implementação que poderia ser melhorado.

O estudo de caso modelado apresenta equações simples atribuídas às atividades com a aplicação de um conjunto reduzido de variáveis. Neste caso, a simulação está refletindo a ideia que o gerente já tem deste mesmo problema. Apesar de não estar descobrindo algo totalmente novo a respeito do sistema, o gerente não conhece os valores exatos para as variáveis nas configurações otimizadas e são estes detalhes que refletem o potencial da solução. A utilização de equações simples para as atividades e processos com a aplicação de um conjunto reduzido de variáveis prejudica também a condução da análise da sensibilidade.

## 6.3 Trabalhos futuros

O desenvolvimento deste modelo de simulador está situado entre a modelagem e a execução do processo de software, inviabilizando a condução de uma simulação contínua. Neste sentido, identifica-se como trabalho futuro a concepção de um modelo no qual a simulação seja conduzida durante a execução do processo, potencializando ajustes pontuais no processo para qualificá-lo.

Também seria interessante comparar os resultados do método de simulação com técnicas já consolidadas e comuns em empresas, técnicas baseadas em conhecimento, a combinação de diferentes técnicas ou com técnicas de estimativa, como *Wideband Delphi* ou mesmo o *Planning Poker*, usada em métodos ágeis. Análise semelhante poderia ser feita para análise de riscos e planejamento de orçamento.

Gerentes de projetos sem conhecimento sobre simulação e análise poderiam obter indicações de riscos, planejar seus cronogramas e orçamentos caso fosse disposta uma base de conhecimento com equações que modelam situações comuns em grandes

empresas, utilizando equações mais complexas para as atividades e processos com a aplicação de um conjunto amplo de variáveis e, conduzindo outras abordagens da análise da sensibilidade.

Dispor de uma avaliação real, baseada em alguns projetos que foram levados desde a concepção até o término do “período de garantia”, também pode ser identificado como trabalho futuro. Medidas como ociosidade, retrabalho, volume das mudanças em virtude das alterações nas especificações, entre outras, poderiam ser consideradas neste estudo.

## REFERÊNCIAS

- BASILI, V. R. **The role of experimentation in software engineering: past, current, and future.** [S.l.]: IEEE Computer Society. 1996. p. 442-449.
- BEECHAM, S.; HALL, T.; RAINER, A. Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis. **Empirical Softw. Engg.**, v. 8, p. 7-42, 2003.
- CANGUSSU, J. W.; DECARLO, R. A.; MATHUR, A. P. Using sensitivity analysis to validate a state variable model of the software test process. **Software Engineering, IEEE Transactions on**, v. 29, p. 430-443, 2003.
- CHINNECK, J. W. Practical Optimization: A Gentle Introduction. [S.l.]: <http://www.sce.carleton.ca/faculty/chinneck/po.html>, 2006. Cap. Sensitive Analysis.
- CURTIS, B. Guest Editor's Introduction: The Global Pursuit of Process Maturity. **IEEE Software**, v. 17, p. 76-78, 2000.
- DANIELSEN, J.; HOFFMAN, D. B. **Simulation of a cold heavy oil production with sand (CHOPS) separation system.** [S.l.]: Society for Computer Simulation International. 2009. p. 1-9.
- DORES, F. A. **Um Modelo de Simulação de Processos de Software Baseado em Conhecimento para o Ambiente PROSOFT.** PPGC da UFRGS. [S.l.]. 2001.
- FUGGETTA, A. **Software process: a roadmap.** [S.l.]: ACM. 2000. p. 25-34.
- FUKS, K.; KAWA, A. **Simulation of Resource Acquisition by e-Sourcing Clusters Using NetLogo Environment.** [S.l.]: Springer-Verlag. 2009. p. 687-696.
- GRUHN, V. Process-Centered Software Engineering Environments, A Brief History and Future Challenges. **Ann. Softw. Eng.**, v. 14, p. 363-382, 2002.
- HEIMANN, P. et al. Dynamite: dynamic task nets for software process management. **18th international conference on Software engineering (ICSE '96)**, IEEE Computer Society, Washington, DC, USA, 1996. 331-341.
- ITHINK, 2011. Disponível em: <http://www.iseesystems.com/software/Business/ithinkSoftware.aspx>. Acesso em: jul 2011.
- KELLNER, M. I.; MADACHY, R. J.; RAFFO, D. M. Software process simulation modeling: Why? What? How? **Journal of Systems and Software**, v. 46, p. 91-105, 1999.

- KIM, S.; SARJOUGHIAN, H. S.; ELAMVAZHUTHI, V. **DEVS-suite**: a simulator supporting visual experimentation design and behavior monitoring. [S.l.]: Society for Computer Simulation International. 2009. p. 1-7.
- KIM, Y. C.; HAM, K. S.; KIM, T. G. **Object-oriented memory management in DEVSIM++**. [S.l.]: ACM. 1993. p. 670-673.
- KIRK, D. **A Flexible Software Process Model**. [S.l.]: IEEE Computer Society. 2004. p. 57-59.
- KITCHENHAM, B. A. et al. Preliminary guidelines for empirical research in software engineering. **IEEE Trans. Softw. Eng.**, v. 28, p. 721-734, 2002.
- KLÜGL, F.; HERRLER, R.; FEHLER, M. **SeSAM**: implementation of agent-based simulation using visual programming. [S.l.]: ACM. 2006. p. 1439-1440.
- KNAPP, V. **The Smalltalk simulation environment**. [S.l.]: ACM. 1986. p. 125-128.
- KRAHL, D. **Extend**: an interactive simulation tool: extend: an interactive simulation tool. [S.l.]: Winter Simulation Conference. 2003. p. 188-196.
- LEHR, W. et al. **Model sensitivity analysis in environmental emergency management**: a case study in oil spill modeling. [S.l.]: [s.n.]. 1994. p. 1198-1205.
- LIEBROCK, L. M. **Empirical sensitivity analysis for computational procedures**. [S.l.]: [s.n.]. 2005. p. 32-35.
- LIMA, C. A. **Uma abordagem flexível para execução de processos de software evolutivos**. Universidade Federal do Rio Grande do Sul - PPGC. [S.l.]. 2003. Disponível em: \$\$.
- LO, J.-H. et al. **Sensitivity analysis of software reliability for component-based software applications**. [S.l.]: [s.n.]. 2003. p. 500-505.
- LONCHAMP, J. **A Structured Conceptual and Terminological Framework for Software Process Engineering**. [S.l.]: New York: IEEE Computer Society. 1993. p. 41-53.
- LU, M.; CHAN, W.-H.; YEUNG, D. S. **Sensitivity analysis on a construction operations simulation model using neural networks**. [S.l.]: [s.n.]. 2005. p. 4173-4178 Vol. 7.
- M., P. H.; VERBRAECK, A.; RENGELINK, W. **Emulation with DSOL**. [S.l.]: Winter Simulation Conference. 2005. p. 1453-1462.
- MÜNCH, J.; ROMBACH, D.; RUS, I. **Creating an Advanced Software Engineering Laboratory by Combining Empirical Studies with Process Simulation**. [S.l.]: IEEE Computer Society. 2003.
- MURTA, L.; BARROS, M.; WERNER, C. Charon: uma ferramenta para a modelagem, simulação, execução e acompanhamento de processos de software. **XVI Simpósio Brasileiro de Engenharia de Software**, Gramado, RS, Brasil, 2002. 366-371.
- NAYLOR, T. H. et al. **Computer Simulation Techniques**. [S.l.]: John Wiley and Sons Ltd., 1966.
- NICOLAU, B. B. **Um Simulador Estocástico de Processo de Software Baseado em Conhecimento**. Universidade Federal do Pará. [S.l.]. 2009.

- NORTH, M. J.; COLLIER, N. T.; VOS, J. R. Experiences creating three implementations of the repast agent modeling toolkit. **ACM Trans. Model. Comput. Simul.**, v. 16, p. 1-25, 2006.
- PARK, S. et al. **Deriving Software Process Simulation Model from SPEM-based Software Process Model**. [S.l.]: IEEE Computer Society. 2007. p. 382-389.
- PERRY, D. E.; PORTER, A. A.; VOTTA, L. G. **Empirical studies of software engineering: a roadmap**. [S.l.]: ACM Press. 2000. p. 345-355.
- PFAHL, D.; RUHE, G.; KOVAL, N. **An Experiment for Evaluating the Effectiveness of Using a System Dynamics Simulation Model in Software Project Management Education**. [S.l.]: IEEE Computer Society. 2001. p. 97.
- PROCESSMODEL, 2011. Disponivel em: <<http://www.processmodel.com/>>. Acesso em: jul 2011.
- PROCESSWEAVER, 2011. Disponivel em: <<http://www.processweaver.com/>>. Acesso em: jul 2011.
- QSIM, 2011. Disponivel em: <<http://www.eecs.umich.edu/~kuipers/research/qsim/qsim-overview.html>>. Acesso em: jul 2011.
- RAFFO, M. I. Empirical analysis in software process simulation modeling. **Journal of Systems and Software**, v. 53, p. 31-41, 2000.
- ROBERTS, C. **Describing and acting process models with PML**. [S.l.]: ACM. 1988. p. 136-141.
- RÖSCH, P. et al. The Spearmint Software Process Modeling Environment. **20th International Conference on Software Engineering, Vol. II, IEEE Computer**, 1998.
- SIGMAFLOW, 2011. Disponivel em: <<http://www.sigmaflow.com/sigmaflow-business-processintelligence.html>>. Acesso em: jul 2011.
- SIMPROCESS, 2011. Disponivel em: <<http://www.simprocess.com/>>. Acesso em: jul 2011.
- SIMSE, 2011. Disponivel em: <<http://www.ics.uci.edu/~emilyo/SimSE/>>. Acesso em: jul 2011.
- VENSIM, 2011. Disponivel em: <<http://www.vensim.com/>>. Acesso em: jul 2011.
- VERAN, M.; POTIER, D. **QNAP 2: A portable environment for queueing systems modelling**. INRIA. [S.l.]. 1984.
- WEBAPSEE, 2011. Disponivel em: <[http://www.ufpa.br/webapsee/index.php?option=com\\_content&view=article&id=47&Itemid=103&lang=br](http://www.ufpa.br/webapsee/index.php?option=com_content&view=article&id=47&Itemid=103&lang=br)>. Acesso em: jul 2011.
- WERNICK, P.; HALL, T. **Getting the Best out of Software Process Simulation and Empirical Research in Software Engineering**. [S.l.]: [s.n.]. 2007. p. 3-3.
- ZHANG, H.; KITCHENHAM, B. A.; PFAHL, D. **Reflections on 10 Years of Software Process Simulation Modeling: A Systematic Review**. [S.l.]: [s.n.]. 2008. p. 345-356.

ZHANG, H.; KITCHENHAM, B.; JEFFERY, R. **Planning Software Project Success with Semi-Quantitative Reasoning**. [S.l.]: [s.n.]. 2007. p. 369-378.

## ANEXO A – RESULTADOS OBTIDOS COM A REVISÃO SISTEMÁTICA DOS TRABALHOS PUBLICADOS SOBRE SIMULAÇÃO DE PROCESSOS DE SOFTWARE

RefBib	Why (purpose)	What (model scope)	How (approache/ language)	When (activities)	Extras				Natureza do artigo
					Special Techniques	Tools	Problem	Year	
CollofelloNov2000	Training or Learning	Development Project	System Dynamics	Project Management		iThink	Project Failures	2000	Tool
Drappa2000a	Training or Learning	Development Project	Hybrid Simulation	Decision Making	Iterative Sim - QAModel	SESAM	Verification Efficiency of PM Learning	2000	Model
Martin2000a	Management	Portion of Life Cycle	Discrete event simulation	Process Modeling		Extend	Potentialities of Hybrid Simulation	2000	Model
Martin2000	Management	Development Project	Discrete event simulation	Project Management			Process Changes	2000	Experience report
Raffo2000	Management	Development Project	System Dynamics	Process Modeling	Monte Carlo			2000	Model
Scacchi2000	Understanding	Portion of Life Cycle	Rule-based languages	Evolution		Loom e Ontosaurus	Redesign, Evolution	2000	Experience report
Withers2000	Improvement	Development Project	Discrete event simulation	Process Modeling			Import Best Practice from SE to Sim	2000	Methodology
Wu2000	Management	Development Project	Rule-based languages	Project Planning	Intelligent Agents	PAMPA2	Project Planning	2000	Tool
Choi2001	Improvement	Portion of Life Cycle	Discrete event simulation	Software Acquisition	PML, HLA, RTI, Web Infrastructure	Self Development	Software Acquisition Coordination	2001	Tool
Donzelli2001b	Management	Development Project	Hybrid Simulation	Process Modeling	SD, De and Analytical Methods	QNAP2	Effects of Requirements instability	2001	Model
Donzelli2001a	Improvement	Portion of Life Cycle	Hybrid Simulation	Decision Making	SD, De and Analytical Methods	QNAP2	Validation of Testing Process	2001	Tool
Gruhn2001	Improvement	Multi-project	State-based process models	Process Modeling	PML		Consistency Checking	2001	Tool
Henderson2001	Improvement	Development Project	State-based process models	Decision Making		RoEnact	Process Improvement	2001	Tool

Host2001	Improvement	Portion of Life Cycle	Discrete event simulation	Requirements Management		REPEAT	Market Requirement Management Process	2001	Model
Houston2001	Understanding	Development Project	System Dynamics	SP Simulation	Sensitivity Analysis, Experimental Design		Sensitivity analysis Assessment	2001	Methodology
Houston2001a	Management	Development Project	System Dynamics	Risk Management	Stochastic Simulation		Risk Management	2001	Model
Kahen2001	Improvement	Long term project	System Dynamics	Software Evolution			Policy Investigation	2001	Experience report
Kim2001	Understanding	Portion of Life Cycle	Hybrid Simulation	Decision Making	Project Management		Project Coordination	2001	Experience report
Mandl-Striegnitz2001	Training or Learning	Development Project	Hybrid Simulation	Decision Making	Interactive Simulation	SESAM	Project Management Experience	2001	Experience report
Martin2001	Management	Development Project	Hybrid Simulation	Project Management		Extend	Project Changes Evaluation	2001	Model
Podnar2001	Management	Portion of Life Cycle	Discrete event simulation	Software Maintenance	Monte Carlo, Poison Process		Evaluate Software Maintenance	2001	Model
Reis2001	Improvement	Development Project	State-based process models	Process Modeling	CBR, PML, Polices	APSEE	Reuse	2001	Model
Ruiz2001	Management	Development Project	System Dynamics	Process Modeling	Reduce Dynamic Model		Incomplete Historical Data	2001	Model
Stallinger2001	Management	Portion of Life Cycle	System Dynamics	Requirements Management	EasyWinWin	Extend	Requirements Definition	2001	Model
StrikeOct2001	Management	Development Project	Discrete event simulation	Project Cost Estimation	Listen Deletion, Mean Imputation, Hot-deck Imputation, Monte Carlo		Incomplete Historical Data	2001	Model
Anjewierden2002	Management	Development Project	Discrete event simulation	Process Modeling	Knowledge Management	KMsim	Appropriate System Modeling and Simulation	2002	Tool
Chiang2002	Decision Support	Portion of Life Cycle	Discrete event simulation	Quality Assurance	Monte Carlo, Machine Learning	Softgoal	Early Lifecycle Evaluation	2002	Tool
Haberlein2002	Improvement	Portion of Life Cycle	Discrete event simulation	Software Acquisition	Interactive Simulation			2002	Tool
Hanakawa2002	Management	Development Project	System Dynamics	Decision Making	Knowledge Management		Prediction of Productivity Variations	2002	Model
Jaafari2002	Management	Development Project	Discrete event simulation	Project Management	Holistic Evaluation	DSMS - Dynamic Simulation Modeling System	Evaluation of Overall Project Functionalities	2002	Tool



Menzies2002a	Management	Development Project	Discrete event simulation	Process Modeling	COCOMO II, Discrete Event Simulation, TAR2 Treatment Learning		Tools Selection	2002	Experience report
Perez2002	Management	Development Project	Discrete event simulation	Process Evaluation	Knowledge Database	RAPIDE	Software Process Evaluation	2002	Experience report
Padberg2002a	Management	Development Project	Discrete event simulation	Project Management	Dynamic Optimization Techniques	Extend	Scheduling Software Projects	2002	Model
Pfahl2002	Management	Development Project	System Dynamics	Process Modeling			Modeling and Simulation Methods	2002	Methodology
Raffo2002	Management	Development Project	Discrete event simulation	Decision Making			Trade-off Analysis	2002	Model
Ramil2002	Understanding	Development Project	System Dynamics	Software Evolution	Qualitative Simulation	QSim	Incomplete Knowledge	2002	Model
Ruiz2002	Improvement	Development Project	System Dynamics	Software Process Improvement		DIFSPI	Software Improvement	2002	Tool
Rus2002	Management	Development Project	Discrete event simulation	V & V	Empirical Data		Decision Making from Personal Experience	2002	Model
Kuppuswami2003	Management	Development Project	System Dynamics	Extreme Programming		Vensim	Analyze the Effects of Individual XP Practices on Development Effort	2003	Model
Munch2003	Management	Development Project	Discrete event simulation	V & V	Empirical Data	Extend	Creation of Accurate Simulation Models	2003	Model
Padberg2003	Management	Development Project	Discrete event simulation	Scheduling	Probabilistic Model	Extend	Scheduling Software Projects	2003	Model
Pfahl2003	Training or Learning	Development Project	System Dynamics	Project Planning	Replicated Experiment		Project Management	2003	Experience report
Raffo2003a	Management	Development Project		Process Modeling	Modularization		Improvement of Simulations Models	2003	Model
Rus2003a	Management	Development Project	Discrete event simulation	Quality Assurance	Empirical Data		Decision Making from Personal Experience	2003	Model
Rus2003	Management	Development Project	Discrete event simulation	Process Modeling	Methodology			2003	Methodology
Wernick2003	Improvement	Development Project					Assessment of Simulation Models	2003	Methodology

Antoniol2004	Management	Development Project	Discrete event simulation	Software Maintenance	Queuing Simulation		Assessing Staffing Needs	2004	Model
Wakeland2004	Management	Development Project	Hybrid Simulation	Inspection	Sensitivity Analyses, Design Experiment,	Extend	Evaluate Processes Changes	2004	Experience report
Carmona2004	Understanding		Discrete event simulation		Generalized Model	DIAMSI M	Development of a Generalized Simulator	2004	Tool
Joslin2005	Management	Development Project	Discrete event simulation	Resource Management	Intelligent Agents		Assist Decision Making	2004	Experience report
Ruiz2004	Management	Development Project	System Dynamics	Process Improvement	Sim e Quality Cost	Vensim	Agile SP	2004	Model
Ruiz2004a	Understanding	Development Project	System Dynamics	COTS Development	Sensitivity Analysis	Vensim	Cots Development	2004	Model
Fernandez2004a	Training or Learning	Development Project	Discrete event simulation	Process Modeling	Role Playing	Spearmint/Xchips	Tecnologic Support	2004	Tool
Haberlein2004	Management	Portion of Life Cycle	System Dynamics	Software Acquisition		Vensim	Software Acquisition Coordination	2004	Model
Chen2004	Management	Development Project	Discrete event simulation	Evolution		DevsJava	Software Product Line	2004	Tool
Birkhölzer2005	Understanding	Development Project	Hybrid Simulation	Strategic Management	Interactive Simulation		Process Change Control	2005	Tool
Navarro2005a	Training or Learning	Development Project	Discrete Event Simulation	Learning	Single Player, Role Playing	SimSE		2005	Tool
Padberg2005	Decision Support	Development Project	Discrete Event Simulation	Project Schedule	Markov Decision Process		Exponential Grow of Simulation Time	2005	Model
Munch2005	Decision Support	Development Project	Hybrid Simulation	Software Development	Virtual Lab Experimentation	Gensim, etc	Trade-off Analysis	2005	Experience report
Smith2005	Understanding	Development Project	System Dynamics	Software Evolution	Qualitative Simulation Techniques		Relation Between Size and Complexity	2005	Model
Choi2006	Management	Development Project	Hybrid Simulation	Software Acquisition		DEVS-based hybrid SPSM (Devsim++)		2006	Experience report
Yilmaz2006	Management	Development Project	Discrete event simulation	Resource Management	Intelligent Agents	Repast Agent Simulation		2006	Tool
Pfahl2006	Management	Portion of Life Cycle	System Dynamics	Software Release Planning	Stability Analyses	REPSIM/Release Planner/Evolve		2006	Model
Stopford2006	Understanding	Portion of Life Cycle	Discrete event simulation	Software Evolution	Agents			2006	Experience report
Zhang2006	Management	Development Project	Hybrid Simulation	Resource Management	Qualitative Simulation	QSim	Quantitative Constraints	2006	Model
Setamanint2006	Management	Development Project	Hybrid Simulation	Project Planning	A New Model		How Simulation Can	2006	Model

							Support GSD		
Hood2006	Training or Learning	Development Project	Discrete event simulation	Learning	Lego	SimProject	Teaching Project Management	2006	Experience report
Noothong2006	Decision Support	Development Project	Discrete event simulation	Project Planning	Bayesian Network Decision Network		Resource Allocation	2006	Model
Huang2006	Management	Development Project	Discrete event simulation	Process Modeling	Petri Nets		Value Based Process Modeling	2006	Model
Madachy2006	Decision Support	Development Project	System Dynamics	Process Modeling	COQUALMO, VBSE		Cost and Time Estimation	2006	Model
Turnu2006	Understanding	Development Project	System Dynamics	Software Development			Effects of Adding TDD in Open Source Development	2006	Model
Zhang2006b	Management	Development Project	Hybrid Simulation	Project Management	Semi-Qualitative Simulation	QSim	Software Staffing Process	2006	Model
Smith2006	Understanding	Development Project	Discrete event simulation	Software Development	Agent-based Model	NetLogo Agent Simulation Tool	Decentralised Development, Size, Complexity and Effort	2006	Tool
Donzelli2006	Decision Support	Development Project	Hybrid Simulation	Software Development	Levels of Abstraction (High and Low levels)	Queuing Networks Analysis Package (QNAP2)	Requirements Instability	2006	Model
Ge2006	Decision Support	Portion of Life Cycle	Discrete Event Simulation	Project Management	Generic Algorithms		Software Schedule	2006	Model
Dai2007	Management	Development Project	Hybrid Simulation	Project Planning	Multi-agent Modelling/ Discrete Event Simulation	Soft-PM and OEC-SPM		2007	Experience report
Naunchan2007	Management	Development Project	System Dynamics	Project Planning	Analytical Techniques		Effort, Time of COTs	2007	Model
Jeng2007	Management	Development Project	System Dynamics	Project Management		Vensim	SOA	2007	Model
Zhang2007b	Decision Support	Development Project	Hybrid Simulation	Adoption of Simulation Paradigm	CMM Levels		Simulation Adoption	2007	Model
Park2007	Decision Support	Development Project	Hybrid Simulation	Acquisition	UML	Devsim++	Simulation Adoption	2007	Model
Zhang2007	Management	Development Project	Hybrid Simulation	Project Management	Semi-Qualitative Simulation	QSim	Evolved Definition of Project Success	2007	Model
Setamanit2007b	Evaluation	Development Project	Hybrid Simulation	Project Planning	A New Model		Allocation Strategy, Project Duration	2007	Experience report
Deleris2007	Management	Development Project	Discrete event simulation	Project Management	Monte Carlo Simulation, Apply Risk	Self Development	Project Schedule	2007	Experience report

Katsamakos2007	Understanding	Development Project	System Dynamics	Software Development	Only a Model	Only a Model	Interaction Between Developers	2007	Model
Zhu2007	Training or Learning	Development Project	Discrete event simulation	Learning	Multiplayer Online Game	SimSE, MO-SEProcess		2007	Experience report
Srinivasan2007	Training or Learning	Development Project	Discrete event simulation	Learning	Constructivism		Teaching Software Process	2007	Experience report
Al-Emran2008	Decision Support	Portion of Life Cycle	Discrete event simulation	Software Release Planning	Heuristics Based Optimization	Extend	Process Management	2008	Methodology
Zhang2008	Improvement	Portion of Life Cycle	Hybrid Simulation	Software Test		Extend	Productivity of Testers	2008	Model
Raffo2008	Decision Support	Portion of Life Cycle	Discrete event simulation	New Technology adoption	What-if Analysis	Extend	Impact of Process Changes	2008	Model

## ANEXO B - DESCRITIVO DAS PRINCIPAIS CLASSES DO SIMULADOR

### 1. Class DataAccess

java.lang.Object

**com.nuovonet.apseesim.connect.DataAccess**

public class **DataAccess**

extends java.lang.Object

Classe responsável pela conexão com a base de dados e com o acesso às informações desta base.

**Author:** Juliano Dertzbacher

Field Summary	
(package private) static java.sql.Connection	<a href="#"><u>db</u></a>
(package private) static <a href="#"><u>Parametros</u></a>	<a href="#"><u>param</u></a>
(package private) static java.sql.Statement	<a href="#"><u>sql</u></a>

Constructor Summary	
<a href="#"><u>DataAccess()</u></a>	

Method Summary	
private static void	<a href="#"><u>closeConn()</u></a> Método que encerra a conexão com a base de dados.
static java.util.List	<a href="#"><u>getActivityConstantTimeData</u></a> (long oid) Método que retorna os dados de tempo básico, nível de habilidade requerido e fator de multiplicação para uma atividade específica.
static <a href="#"><u>Activity</u></a>	<a href="#"><u>getActivityData</u></a> (long oid) Método que retorna dados de uma atividade.
static java.util.List	<a href="#"><u>getActivityData</u></a> ( <a href="#"><u>Process</u></a> prc) Método que retorna dados das atividades de um

	processo.
static long	<a href="#"><u>getActivitySequenceOID(Activity act)</u></a> Método que retorna a atividade de sequência para uma atividade informada.
static java.util.List	<a href="#"><u>getAgentAbilityData(Agent ag)</u></a> Método que retorna os dados de habilidade de um agente.
static java.util.List	<a href="#"><u>getAgentData(Group grp)</u></a> Método que retorna os dados dos agentes de um grupo.
static java.util.List	<a href="#"><u>getAllProcessData(long oid)</u></a> Método que retorna informações de todos os processos.
static java.util.List	<a href="#"><u>getArtifactINData(Activity act)</u></a> Método que retorna os artefatos de entrada de uma atividade.
static java.util.List	<a href="#"><u>getArtifactOUTData(Activity act)</u></a> Método que retorna os artefatos de saída de uma atividade.
static long	<a href="#"><u>getConcurrentActivity(long destAct, long act)</u></a> Método que retorna o identificador da atividade concorrente a uma atividade específica.
private static void	<a href="#"><u>getConnection()</u></a> Método que abre e disponibiliza a conexão à base de dados.
static java.util.List	<a href="#"><u>getGroupData(long oid)</u></a> Método que retorna os dados de um grupo de trabalho.
static java.util.List	<a href="#"><u>getInvolvedGroupsData(Activity act)</u></a> Método que retorna os grupos envolvidos em uma atividade.
static <a href="#"><u>Process</u></a>	<a href="#"><u>getProcessData(long oid)</u></a> Método que acessa e retorna as informações do processo.
private static java.sql.Statement	<a href="#"><u>openStatement()</u></a> Método que abre o Statement e o retorna para o uso das consultas à base de dados.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### 1.1. param

static [Parametros](#) param

---

## 1.2. db

static java.sql.Connection **db**

---

## 1.3. sql

static java.sql.Statement **sql**

### Constructor Detail

## 1.4. DataAccess

public **DataAccess**()

### Method Detail

## 1.5. openStatement

private static java.sql.Statement **openStatement**()

Método que abre o Statement e o retorna para o uso das consultas à base de dados.

**Returns:**

Statement

---

## 1.6. getConnection

private static void **getConnection**()

throws java.lang.ClassNotFoundException,  
java.sql.SQLException

Método que abre e disponibiliza a conexão à base de dados.

**Throws:**

java.lang.ClassNotFoundException  
java.sql.SQLException

---

## 1.7. closeConn

private static void **closeConn**()

Método que encerra a conexão com a base de dados.

---

## 1.8. getProcessData

public static [Process](#) **getProcessData**(long oid)

Método que acessa e retorna as informações do processo.

**Parameters:**

oid - - Identificador do processo a ser obtido (long).

**Returns:**

prc - Classe Process com os atributos carregados.

---

## 1.9. getAllProcessData

public static java.util.List **getAllProcessData**(long oid)

Método que retorna informações de todos os processos.

**Parameters:**

oid - - Identificador do processo a ser obtido(long). Se for preenchido retornará os dados respectivos, caso seja zero (0) serão retornados dados de todos os processos.

**Returns:**

toReturn - List contendo os dados dos processos disponíveis (Process).

---

**1.10. getActivityData**

public static java.util.List **getActivityData**([Process](#) prc)

Método que retorna dados das atividades de um processo.

**Parameters:**

prc - - Processo cujas atividades serão retornadas. Caso seja nulo serão retornadas todas as atividades de todos os processos.

**Returns:**

toReturn - List contendo os dados das atividades (Activity).

---

**1.11. getActivityData**

public static [Activity](#) **getActivityData**(long oid)

Método que retorna dados de uma atividade.

**Parameters:**

oid - - Identificador da atividade cujos dados serao retornados(long).

**Returns:**

toReturn - List contendo os dados das atividades (Activity).

---

**1.12. getConcurrentActivity**

public static long **getConcurrentActivity**(long destAct,  
long act)

Método que retorna o identificador da atividade concorrente a uma atividade específica.

**Parameters:**

destAct - - Identificador da atividade da sequência (long).

act - - Identificador da atividade atual (long).

**Returns:**

toReturn - Identificador da atividade concorrente (long).

---

**1.13. getArtifactOUTData**

public static java.util.List **getArtifactOUTData**([Activity](#) act)

Método que retorna os artefatos de saída de uma atividade.

**Parameters:**

act - - Atividade da qual serao buscados os artefatos de saída.

**Returns:**

toReturn - List com os dados dos artefatos (Artifact).

---

**1.14. getArtifactINData**

public static java.util.List **getArtifactINData**([Activity](#) act)

Método que retorna os artefatos de entrada de uma atividade.

**Parameters:**

act - - Atividade da qual serão buscados os artefatos de entrada.

**Returns:**

toReturn - List com os dados dos artefatos (Artifact).

---



### 1.15. **getGroupData**

public static java.util.List **getGroupData**(long oid)

Método que retorna os dados de um grupo de trabalho.

**Parameters:**

oid - - Identificador do grupo cujos dados serão retornados (long).

**Returns:**

toReturn - List com os dados carregados (Group).

---

### 1.16. **getAgentData**

public static java.util.List **getAgentData**([Group](#) grp)

Método que retorna os dados dos agentes de um grupo.

**Parameters:**

grp - - Grupo cujos agentes serão retornados (Group).

**Returns:**

toReturn - List com os dados dos agentes (Agent).

---

### 1.17. **getAgentAbilityData**

public static java.util.List **getAgentAbilityData**([Agent](#) ag)

Método que retorna os dados de habilidade de um agente.

**Parameters:**

ag - - Agente cuja habilidade será buscada (Agent).

**Returns:**

toReturn - List com as habilidades (Ability).

---

### 1.18. **getInvolvedGroupsData**

public static java.util.List **getInvolvedGroupsData**([Activity](#) act)

Método que retorna os grupos envolvidos em uma atividade.

**Parameters:**

act - - Atividades cujos grupos serão buscados (Activity).

**Returns:**

toReturn - List com os grupos envolvidos (Group).

---

### 1.19. **getActivityConstantTimeData**

public static java.util.List **getActivityConstantTimeData**(long oid)

Método que retorna os dados de tempo básico, nível de habilidade requerido e fator de multiplicação para uma atividade específica.

**Parameters:**

oid - - Identificador da atividade a ser consultada (long).

**Returns:**

toReturn - List com os dados a retornar (ActivityConstantTime).

---

### 1.20. **getActivitySequenceOID**

public static long **getActivitySequenceOID**([Activity](#) act)

Método que retorna a atividade de sequência para uma atividade informada.

**Parameters:**

act - - Atividade de origem (Activity).

**Returns:**

toReturn - Identificador da atividade sequencial (long).

## 2. Class Parametros

java.lang.Object

**com.nuovonet.apseesim.connect.Parametros**

public class **Parametros**

extends java.lang.Object

Classe que fornece os parâmetros do sistema, lidos do arquivo de propriedades.

**Author:** Juliano Dertzbacher

Field Summary	
private static java.lang.String	<a href="#"><u>ARQUIVO_PROPIEDADES</u></a>
private java.lang.String	<a href="#"><u>dbname</u></a>
private java.lang.String	<a href="#"><u>driverClass</u></a>
private static <a href="#"><u>Parametros</u></a>	<a href="#"><u>instancia</u></a>
private java.lang.String	<a href="#"><u>ip</u></a>
private java.lang.String	<a href="#"><u>passwd</u></a>
private java.lang.String	<a href="#"><u>port</u></a>
private java.lang.String	<a href="#"><u>username</u></a>

Constructor Summary	
<a href="#"><u>Parametros()</u></a>	Construtor que seta os padrões lidos do arquivo de propriedades.

Method Summary	
private void	<a href="#"><u>doSetPadrao()</u></a> Método que faz a leitura e popula os atributos da classe, vindos do arquivo de propriedades.
java.lang.String	<a href="#"><u>getDbname()</u></a> Método que retorna o dbname.
java.lang.String	<a href="#"><u>getDriverClass()</u></a> Método que retorna a driverclass.
static <a href="#"><u>Parametros</u></a>	<a href="#"><u>getInstancia()</u></a> Método sincronizado que retorna (ou cria e retorna) a instância

	desta classe.
java.lang.String	<a href="#"><u>getIp()</u></a> Método que retorna o ip.
java.lang.String	<a href="#"><u>getPasswd()</u></a> Método que retorna a passwd.
java.lang.String	<a href="#"><u>getPort()</u></a> Método que retorna a port.
java.lang.String	<a href="#"><u>getUsername()</u></a> Método que retorna o username.
void	<a href="#"><u>setDbname()</u></a> (java.lang.String dbname) Método que seta o dbname.
void	<a href="#"><u>setDriverClass()</u></a> (java.lang.String driverClass) Método que seta driverclass.
void	<a href="#"><u>setIp()</u></a> (java.lang.String ip) Método que seta o ip.
void	<a href="#"><u>setPasswd()</u></a> (java.lang.String passwd) Método que seta a passwd.
void	<a href="#"><u>setPort()</u></a> (java.lang.String port) Método que seta a porta do banco de dados.
void	<a href="#"><u>setUsername()</u></a> (java.lang.String username) Método que seta o username.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### 2.1. instancia

private static [Parametros](#) **instancia**

---

#### 2.2. ARQUIVO\_PROPRIEDADES

private static final java.lang.String **ARQUIVO\_PROPRIEDADES**

See Also:

[Constant Field Values](#)

---

#### 2.3. dbname

private java.lang.String **dbname**

---

#### 2.4. username

private java.lang.String **username**

---

**2.5. passwd**

```
private java.lang.String passwd
```

---

**2.6. ip**

```
private java.lang.String ip
```

---

**2.7. port**

```
private java.lang.String port
```

---

**2.8. driverClass**

```
private java.lang.String driverClass
```

**Constructor Detail****2.9. Parametros**

```
public Parametros()
```

Construtor que seta os padrões lidos do arquivo de propriedades.

**Method Detail****2.10. getInstancia**

```
public static Parametros getInstancia()
```

Método sincronizado que retorna (ou cria e retorna) a instância desta classe.

**Returns:**

instancia do objeto.

---

**2.11. doSetPadrao**

```
private void doSetPadrao()
```

Método que faz a leitura e popula os atributos da classe, vindos do arquivo de propriedades.

---

**2.12. getDbname**

```
public java.lang.String getDbname()
```

Método que retorna o dbname.

**Returns:**

dbname

---

**2.13. setDbname**

```
public void setDbname(java.lang.String dbname)
```

Método que seta o dbname.

**Parameters:**

dbname - nome da base de dados.

---

**2.14. getIp**

```
public java.lang.String getIp()
```

Método que retorna o ip.

**Returns:**

ip

---

**2.15. setIp**

public void **setIp**(java.lang.String ip)

Método que seta o ip.

**Parameters:**

ip - ip da base de dados.

---

**2.16. getPasswd**

public java.lang.String **getPasswd**()

Método que retorna a passwd.

**Returns:**

passwd

---

**2.17. setPasswd**

public void **setPasswd**(java.lang.String passwd)

Método que seta a passwd.

**Parameters:**

passwd - senha da base de dados.

---

**2.18. getUsername**

public java.lang.String **getUsername**()

Método que retorna o username.

**Returns:**

username

---

**2.19. setUsername**

public void **setUsername**(java.lang.String username)

Método que seta o username.

**Parameters:**

username - nome do usuario da base de dados.

---

**2.20. getPort**

public java.lang.String **getPort**()

Método que retorna a port.

**Returns:**

port

---

**2.21. setPort**

public void **setPort**(java.lang.String port)

Método que seta a porta do banco de dados.

**Parameters:**

port - porta do banco de dados

---

**2.22. getDriverClass**

public java.lang.String **getDriverClass**()

Método que retorna a driverclass.

**Returns:**  
driverclass

---

### **2.23. setDriverClass**

public void **setDriverClass**(java.lang.String driverClass)

Método que seta driverclass.

**Parameters:**

driverClass - nome da classe do driver de conexao.

### 3. Class MonteCarlo

java.lang.Object

**com.nuovonet.apseesim.core.MonteCarlo**

public class **MonteCarlo**

extends java.lang.Object

Classe que carrega os parâmetros da simulação Monte Carlo.

**Author:** Juliano Dertzbacher

Field Summary	
private static java.lang.String	<a href="#"><u>ARQUIVO PROPRIEDADES</u></a>
private static <a href="#"><u>MonteCarlo</u></a>	<a href="#"><u>instancia</u></a>
private java.lang.String	<a href="#"><u>iterations</u></a>
private java.lang.String	<a href="#"><u>mha</u></a>
private java.lang.String	<a href="#"><u>mhamax</u></a>
private java.lang.String	<a href="#"><u>mhamin</u></a>
private java.lang.String	<a href="#"><u>qag</u></a>
private java.lang.String	<a href="#"><u>qagmax</u></a>
private java.lang.String	<a href="#"><u>qagmin</u></a>
private java.lang.String	<a href="#"><u>sch</u></a>
private java.lang.String	<a href="#"><u>schmax</u></a>
private java.lang.String	<a href="#"><u>schmin</u></a>

Constructor Summary
<a href="#"><u>MonteCarlo()</u></a> Construtor que seta os padrões lidos do arquivo de propriedades.

Method Summary
----------------



private void	<a href="#"><u>doSetPadrao()</u></a> Método que faz a leitura e popula os atributos da classe, vindos do arquivo de propriedades.
static <a href="#"><u>MonteCarlo</u></a>	<a href="#"><u>getInstancia()</u></a> Método sincronizado que retorna (ou cria e retorna) a instância desta classe.
java.lang.String	<a href="#"><u>getIterations()</u></a> Método que retorna o iterations.
java.lang.String	<a href="#"><u>getMha()</u></a> Método que retorna o mha.
java.lang.String	<a href="#"><u>getMhamax()</u></a> Método que retorna o mhamax.
java.lang.String	<a href="#"><u>getMhamin()</u></a> Método que retorna o mhamin.
java.lang.String	<a href="#"><u>getQag()</u></a> Método que retorna o qag.
java.lang.String	<a href="#"><u>getQagmax()</u></a> Método que retorna o qagmax.
java.lang.String	<a href="#"><u>getQagmin()</u></a> Método que retorna o qagmin.
java.lang.String	<a href="#"><u>getSch()</u></a> Método que retorna o sch.
java.lang.String	<a href="#"><u>getSchmax()</u></a> Método que retorna o schmax.
java.lang.String	<a href="#"><u>getSchmin()</u></a> Método que retorna o schmin.
void	<a href="#"><u>setIterations()</u></a> (java.lang.String iterations) Método que seta o iterations.
void	<a href="#"><u>setMha()</u></a> (java.lang.String mha) Método que seta o mha.
void	<a href="#"><u>setMhamax()</u></a> (java.lang.String mhamax) Método que seta o mhamax.
void	<a href="#"><u>setMhamin()</u></a> (java.lang.String mhamin) Método que seta o mhamin.
void	<a href="#"><u>setQag()</u></a> (java.lang.String qag) Método que seta o qag.
void	<a href="#"><u>setQagmax()</u></a> (java.lang.String qagmax) Método que seta o qagmax.
void	<a href="#"><u>setQagmin()</u></a> (java.lang.String qagmin) Método que seta o qagmin.
void	<a href="#"><u>setSch()</u></a> (java.lang.String sch)

	Método que seta o sch.
void	<a href="#">setSchmax</a> (java.lang.String schmax) Método que seta o schmax.
void	<a href="#">setSchmin</a> (java.lang.String schmin) Método que seta o schmin.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### 3.1. instancia

private static [MonteCarlo](#) **instancia**

---

#### 3.2. ARQUIVO\_PROPRIEDADES

private static final java.lang.String **ARQUIVO\_PROPRIEDADES**

See Also:

[Constant Field Values](#)

---

#### 3.3. qag

private java.lang.String **qag**

---

#### 3.4. qagmin

private java.lang.String **qagmin**

---

#### 3.5. qagmax

private java.lang.String **qagmax**

---

#### 3.6. sch

private java.lang.String **sch**

---

#### 3.7. schmin

private java.lang.String **schmin**

---

#### 3.8. schmax

private java.lang.String **schmax**

---

#### 3.9. mha

private java.lang.String **mha**

---

#### 3.10. mhamin

private java.lang.String **mhamin**

---

### 3.11. mhamax

private java.lang.String **mhamax**

---

### 3.12. iterations

private java.lang.String **iterations**

#### Constructor Detail

### 3.13. MonteCarlo

public **MonteCarlo**()

Construtor que seta os padrões lidos do arquivo de propriedades.

#### Method Detail

### 3.14. getInstancia

public static [MonteCarlo](#) **getInstancia**()

Método sincronizado que retorna (ou cria e retorna) a instância desta classe.

**Returns:**

Instância do objeto.

---

### 3.15. doSetPadrao

private void **doSetPadrao**()

Método que faz a leitura e popula os atributos da classe, vindos do arquivo de propriedades.

---

### 3.16. getMha

public java.lang.String **getMha**()

Método que retorna o mha.

**Returns:**

mha

---

### 3.17. setMha

public void **setMha**(java.lang.String mha)

Método que seta o mha.

**Parameters:**

mha - Média de Habilidade dos Agentes.

---

### 3.18. getMhamax

public java.lang.String **getMhamax**()

Método que retorna o mhamax.

**Returns:**

mhamax.

---

### 3.19. setMhamax

public void **setMhamax**(java.lang.String mhamax)

Método que seta o mhamax.

**Parameters:**

mhamax - Média de Habilidade dos Agentes (máximo).

---

### 3.20. **getMhamin**

public java.lang.String **getMhamin**()  
Método que retorna o mhamin.  
**Returns:**  
mhamin.

---

### 3.21. **setMhamin**

public void **setMhamin**(java.lang.String mhamin)  
Método que seta o mhamin.  
**Parameters:**  
mhamin - Média de Habilidade dos Agentes (mínimo).

---

### 3.22. **getQag**

public java.lang.String **getQag**()  
Método que retorna o qag.  
**Returns:**  
qag.

---

### 3.23. **setQag**

public void **setQag**(java.lang.String qag)  
Método que seta o qag.  
**Parameters:**  
qag - Quantidade de Agentes.

---

### 3.24. **getQagmax**

public java.lang.String **getQagmax**()  
Método que retorna o qagmax.  
**Returns:**  
qagmax

---

### 3.25. **setQagmax**

public void **setQagmax**(java.lang.String qagmax)  
Método que seta o qagmax.  
**Parameters:**  
qagmax - Quantidade de Agentes (máximo).

---

### 3.26. **getQagmin**

public java.lang.String **getQagmin**()  
Método que retorna o qagmin.  
**Returns:**  
qagmin.

---

### 3.27. **setQagmin**

public void **setQagmin**(java.lang.String qagmin)

Método que seta o qagmin.

**Parameters:**

qagmin - Quantidade de Agentes (mínimo).

---

**3.28. getSch**

public java.lang.String **getSch**()

Método que retorna o sch.

**Returns:**

sch.

---

**3.29. setSch**

public void **setSch**(java.lang.String sch)

Método que seta o sch.

**Parameters:**

sch - Somatório do custo-hora.

---

**3.30. getSchmax**

public java.lang.String **getSchmax**()

Método que retorna o schmax.

**Returns:**

schmax

---

**3.31. setSchmax**

public void **setSchmax**(java.lang.String schmax)

Método que seta o schmax.

**Parameters:**

schmax - Somatório do custo-hora (máximo).

---

**3.32. getSchmin**

public java.lang.String **getSchmin**()

Método que retorna o schmin.

**Returns:**

schmin.

---

**3.33. setSchmin**

public void **setSchmin**(java.lang.String schmin)

Método que seta o schmin.

**Parameters:**

schmin - Somatório do custo-hora (mínimo).

---

**3.34. getIterations**

public java.lang.String **getIterations**()

Método que retorna o iterations.

**Returns:**

iterations.

---

**3.35. setIterations**

public void **setIterations**(java.lang.String iterations)

Método que seta o iterations.

**Parameters:**

iterations - Quantidade de iterações da simulação.

## 4. Class Simulate

java.lang.Object

**com.nuovonet.apseesim.core.Simulate**

public class **Simulate**

extends java.lang.Object

Classe que realiza as simulações dos processos.

**Author:** Juliano Dertzbacher

Field Summary	
private static <a href="#">MonteCarlo</a>	<a href="#">mc</a>

Constructor Summary	
<a href="#">Simulate()</a>	

Method Summary	
private static java.util.List	<a href="#">executeProcess</a> ( <a href="#">Process</a> prc) Método que executa a simulação do processo.
private static java.util.List	<a href="#">executeProcessRAND</a> ( <a href="#">Process</a> prc) Método que executa a simulação randômica (Monte Carlo) do processo.
private static double	<a href="#">getMhaDeff</a> (double theValue, boolean mha) Método que retorna o fator para mha.
private static long	<a href="#">getRandomLong</a> (int aStart, int aEnd, java.util.Random aRandom) Método que retorna um número randômico (long).
private static double	<a href="#">getSchDeff</a> (double theValue, boolean sch) Método que retorna o fator para sch.
static long	<a href="#">numberOfIterations</a> () Método que busca o número de iterações para a simulação Monte Carlo.
static <a href="#">Process</a>	<a href="#">returnRandom</a> (long oid) Metodo que retorna a simulação randômica (Monte Carlo) de um processo.
static java.util. List	<a href="#">returnSimulation</a> () Método que retorna a simulação para os processos.
static <a href="#">Process</a>	<a href="#">returnSimulation</a> (long oid) Método que retorna a simulação de um processo.

### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Field Detail

#### 4.1. `mc`

private static [MonteCarlo](#) `mc`

### Constructor Detail

#### 4.2. `Simulate`

public `Simulate()`

### Method Detail

#### 4.3. `numberOfIterations`

public static long `numberOfIterations()`

Método que busca o número de iterações para a simulação Monte Carlo.

**Returns:**

toReturn - Número de iteracoes (long).

---

#### 4.4. `returnSimulation`

public static java.util.List `returnSimulation()`

Método que retorna a simulação para os processos.

**Returns:**

toReturn - List com os processos e resultados (Process).

---

#### 4.5. `returnSimulation`

public static [Process](#) `returnSimulation(long oid)`

Método que retorna a simulação de um processo.

**Parameters:**

oid - - Identificador do processo a ser simulado.

**Returns:**

prc - Processo simulado (Process).

---

#### 4.6. `returnRandom`

public static [Process](#) `returnRandom(long oid)`

Método que retorna a simulação randômica (Monte Carlo) de um processo.

**Parameters:**

oid - - Identificador do processo a ser simulado.

**Returns:**

prc - Processo simulado (Process).

---

#### 4.7. `executeProcess`

private static java.util.List `executeProcess(Process prc)`

Método que executa a simulação do processo.

**Parameters:**

prc - - Processo a ser simulado.



**Returns:**

results - List com os resultados da simulação (Result).

---

**4.8. executeProcessRAND**

```
private static java.util.List executeProcessRAND(Process prc)
```

Método que executa a simulação randômica (Monte Carlo) do processo.

**Parameters:**

prc - - Processo a ser simulado.

**Returns:**

results - List com os resultados da simulação (Result).

---

**4.9. getRandomLong**

```
private static long getRandomLong(int aStart,
    int aEnd,
    java.util.Random aRandom)
```

Método que retorna um número randômico (long).

**Parameters:**

aStart - - Inteiro de início.

aEnd - - Inteiro de fim.

aRandom - - Instância de Random.

**Returns:**

randomnumber - Long de resultado.

---

**4.10. getSchDeff**

```
private static double getSchDeff(double theValue,
    boolean sch)
```

Método que retorna o fator para sch.

**Parameters:**

theValue - - Valor double-precision base.

sch - - Booleano que indica se será calculado o fator.

**Returns:**

value - Double de resultado.

---

**4.11. getMhaDeff**

```
private static double getMhaDeff(double theValue,
    boolean mha)
```

Método que retorna o fator para mha.

**Parameters:**

theValue - - Valor double-precision base.

mha - - Booleano que indica se será calculado o fator.

**Returns:**

value - Double de resultado.

## 5. Class Result

java.lang.Object

**com.nuovonet.apseesim.model.Result**

public class **Result**

extends java.lang.Object

Classe de entidade, referente ao resultado da simulação.

**Author:** Juliano Dertzbacher

Field Summary	
private java.lang.String	<a href="#"><u>atividade</u></a>
private double	<a href="#"><u>custo_total</u></a>
private double	<a href="#"><u>mha</u></a>
private long	<a href="#"><u>qae</u></a>
private long	<a href="#"><u>qag</u></a>
private long	<a href="#"><u>qas</u></a>
private double	<a href="#"><u>sch</u></a>
private double	<a href="#"><u>tempo_total</u></a>

Constructor Summary	
<a href="#"><u>Result()</u></a>	

Method Summary	
java.lang.String	<a href="#"><u>getAtividade()</u></a>
double	<a href="#"><u>getCusto_total()</u></a>
double	<a href="#"><u>getMha()</u></a>
long	<a href="#"><u>getQae()</u></a>

long	<a href="#"><u>getQag()</u></a>
long	<a href="#"><u>getQas()</u></a>
double	<a href="#"><u>getSch()</u></a>
double	<a href="#"><u>getTempo_total()</u></a>
void	<a href="#"><u>setAtividade</u></a> (java.lang.String atividade)
void	<a href="#"><u>setCusto_total</u></a> (double custo_total)
void	<a href="#"><u>setMha</u></a> (double mha)
void	<a href="#"><u>setQae</u></a> (long qae)
void	<a href="#"><u>setQag</u></a> (long qag)
void	<a href="#"><u>setQas</u></a> (long qas)
void	<a href="#"><u>setSch</u></a> (double sch)
void	<a href="#"><u>setTempo_total</u></a> (double tempo_total)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Field Detail

##### 5.1. atividade

private java.lang.String **atividade**

---

##### 5.2. qag

private long **qag**

---

##### 5.3. sch

private double **sch**

---

##### 5.4. mha

private double **mha**

---

**5.5. qae**

private long **qae**

---

**5.6. qas**

private long **qas**

---

**5.7. tempo\_total**

private double **tempo\_total**

---

**5.8. custo\_total**

private double **custo\_total**

**Constructor Detail****5.9. Result**

public **Result()**

**Method Detail****5.10. getCusto\_total**

public double **getCusto\_total()**

---

**5.11. setCusto\_total**

public void **setCusto\_total**(double custo\_total)

---

**5.12. getMha**

public double **getMha()**

---

**5.13. setMha**

public void **setMha**(double mha)

---

**5.14. getQae**

public long **getQae()**

---

**5.15. setQae**

public void **setQae**(long qae)

---

**5.16. getQag**

public long **getQag()**

---

**5.17. setQag**

public void **setQag**(long qag)

---

**5.18. getQas**

public long **getQas()**

---

**5.19. setQas**

public void **setQas**(long qas)

---

**5.20. getSch**

public double **getSch**()

---

**5.21. setSch**

public void **setSch**(double sch)

---

**5.22. getTempo\_total**

public double **getTempo\_total**()

---

**5.23. setTempo\_total**

public void **setTempo\_total**(double tempo\_total)

---

**5.24. getAtividade**

public java.lang.String **getAtividade**()

---

**5.25. setAtividade**

public void **setAtividade**(java.lang.String atividade)

## 6. Class Exporter

java.lang.Object

**com.nuovonet.apseesim.util.Exporter**

public class **Exporter**  
 extends java.lang.Object  
 Classe utilitária para a exportação de arquivo CSV.  
**Author:** Juliano Dertzbacher

### Constructor Summary

[Exporter\(\)](#)

### Method Summary

static void	<a href="#">enviaArquivo</a> (javax.servlet.http.HttpServletResponse res, java.io.File arquivo) Método que realiza o envio do arquivo de exportação para o navegador.
static java.io. File	<a href="#">exportCSV</a> (java.util.List results) Método que realiza a exportação dos resultados para o arquivo.
static java.io. File	<a href="#">exportRandomicCSV</a> (java.util.List results) Método que realiza a exportação dos resultados randômicos, com todas as iterações do Monte Carlo para um mesmo arquivo.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### 6.1. Exporter

public **Exporter**()

### Method Detail

#### 6.2. exportCSV

public static java.io.File **exportCSV**(java.util.List results)  
 Método que realiza a exportação dos resultados para o arquivo.  
**Parameters:**  
 results - - List dos resultados da simulação (Result).  
**Returns:**  
 result - Arquivo CSV a ser exportado.

#### 6.3. exportRandomicCSV

public static java.io.File **exportRandomicCSV**(java.util.List results)

Método que realiza a exportação dos resultados randômicos, com todas as iterações do Monte Carlo para um mesmo arquivo.

**Parameters:**

results - - List dos resultados da simulação (Result).

**Returns:**

result - Arquivo CSV a ser exportado.

---

#### 6.4. enviaArquivo

```
public static void enviaArquivo(javax.servlet.http.HttpServletResponse res,  
    java.io.File arquivo)  
    throws java.lang.Exception
```

Método que realiza o envio do arquivo de exportação para o navegador.

**Parameters:**

res - - HttpServletResponse que proverá o OutputStream.

arquivo - - File a ser enviado.

**Throws:**

java.lang.Exception