

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA**

ANDRÉ AUGUSTO TISSOT

**INFLUÊNCIA DA REVISÃO DE ATIVIDADES EXECUTADAS PARA
MELHORIA DA ACURÁCIA NA ESTIMATIVA DE SOFTWARE
UTILIZANDO *PLANNING POKER***

DISSERTAÇÃO

**CURITIBA
2015**

ANDRÉ AUGUSTO TISSOT

**INFLUÊNCIA DA REVISÃO DE ATIVIDADES EXECUTADAS PARA
MELHORIA DA ACURÁCIA NA ESTIMATIVA DE SOFTWARE
UTILIZANDO *PLANNING POKER***

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre, do Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná.
Área de Concentração: Engenharia de Software.

Orientador: Prof. Dr. Laudelino Cordeiro Bastos

Coorientadora: Prof. Dr. Maria Cláudia Figueiredo Pereira Emer

CURITIBA

2015

Dados Internacionais de Catalogação na Publicação

T616i
2015 Tissot, André Augusto
Influência da revisão de atividades executadas para melhoria da acurácia utilizando planning poker / André Augusto Tissot.-- 2015.
186 f. : il.; 30 cm

Texto em português, com resumo em inglês
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Computação Aplicada, Curitiba, 2015
Bibliografia: p. 132-138

1. Medição de software. 2. Gerenciamento de configurações de software. 3. Software - Desenvolvimento. 4. Software - Testes. 5. Computação - Dissertações. I. Bastos, Laudelino Cordeiro, orient. II. Emer, Maria Cláudia Figueiredo Pereira, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Computação Aplicada. IV. Título.

CDD: Ed. 22 -- 621.39

Biblioteca Central da UTFPR, Câmpus Curitiba

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO Nº 32

Aos 21 dias do mês de agosto de 2015 realizou-se na sala C-301 a sessão pública de Defesa da Dissertação de Mestrado intitulada "Influência da Revisão de Atividades Executadas para Melhoria da Acurácia na Estimativa de Software Utilizando Planning Poker", apresentada pelo aluno **André Augusto Tissot** como requisito parcial para a obtenção do título de Mestre em Computação Aplicada, na área de concentração "Engenharia de Sistemas Computacionais", linha de pesquisa "Engenharia de Software".

Constituição da Banca Examinadora:

Prof. Dr. Laudelino Cordeiro Bastos, UTFPR - CT (Presidente) _____

Prof. Dr. Adolfo Gustavo Serra Seca Neto, UTFPR - CT _____

Prof. Dr. Andrey Ricardo Pimentel, UFPR _____

Prof^a Dr^a Nádia Puchalski Kozievitch, UTFPR, CT _____

Em conformidade com os regulamentos do Programa de Pós-Graduação em Computação aplicada e da Universidade Tecnológica Federal do Paraná, o trabalho apresentado foi considerado _____ (aprovado/reprovado) pela banca examinadora. No caso de aprovação, a mesma está condicionada ao cumprimento integral das exigências da banca examinadora, registradas no verso desta ata, da entrega da versão final da dissertação em conformidade com as normas da UTFPR e da entrega da documentação necessária à elaboração do diploma, em até _____ dias desta data.

Ciente (assinatura do aluno): _____

(para uso da coordenação)

A Coordenação do PPGCA/UTFPR declara que foram cumpridos todos os requisitos exigidos pelo programa para a obtenção do título de Mestre.

Curitiba PR, ____/____/____

"A Ata de Defesa original está arquivada na Secretaria do PPGCA".

RESUMO

TISSOT, André Augusto. Influência da Revisão de Atividades Executadas para Melhoria da Acurácia Utilizando *Planning Poker*. 2015. 186 f. Dissertação – Programa de Pós-Graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Introdução – A área de pesquisa de estimativa de esforço de software busca melhorar a acurácia das estimativas de projetos e atividades de software.

Objetivo – Este trabalho descreve o desenvolvimento e uso de uma ferramenta web de coleta de dados gerados durante a execução da técnica de estimativa *Planning Poker* e a análise dos dados coletados para investigação do impacto da revisão de dados históricos de esforço.

Método – Foram realizadas estimativas com e sem revisão, em experimentos com alunos de computação da Universidade Tecnológica Federal do Paraná, coletando os dados relacionados à tomada de decisão em uma ferramenta web. Após isso, foi analisado o impacto causado pelas revisões na acurácia da estimativa de esforço de software utilizando *Planning Poker*.

Resultados Obtidos – Foi analisado o comportamento de 14 grupos de estimativas. Dentre esses times, 8 deles tiveram uma melhora na acurácia maior que 50% das estimativas analisadas. Em 3 deles, a soma das estimativas que tiveram melhora com as estimativas que permaneceram estáveis ultrapassou os 50%. Em apenas 3 deles, as estimativas tiveram redução de acurácia maior que 50%.

Conclusões – A Revisão de Atividades Executadas, utilizando *Planning Poker*, melhorou a estimativa de esforço na maioria dos casos analisados, podendo ser um importante método para aprimorar o processo de desenvolvimento de software.

Palavras-chave: Métricas de Software, Predição de Esforço de Software, Estimativa de Esforço de Software, Predição de Custo de Software, Estimativa de Custo de Software, Gerência de Processo de Software.

ABSTRACT

TISSOT, André Augusto. Influence of the Reviewing of Executed Activities to Improve Accuracy Using Planning Poker. 2015. 186 f. Dissertação – Programa de Pós-Graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná. Curitiba, 2015.

Abstract – Background – The software effort estimation research area aims to improve the accuracy of this estimation in software projects and activities.

Aims – This study describes the development and usage of a web application to collect data generated from the Planning Poker estimation process and the analysis of the collected data to investigate the impact of revising previous estimates when conducting similar estimates in a Planning Poker context.

Method – Software activities were estimated by Universidade Tecnológica Federal do Paraná (UTFPR) computer students, using Planning Poker, with and without revising previous similar activities, storing data regarding the decision-making process. And the collected data was used to investigate the impact that revising similar executed activities have in the software effort estimates' accuracy.

Obtained Results – The UTFPR computer students were divided into 14 groups. Eight of them showed accuracy increase in more than half of their estimates. Three of them had almost the same accuracy in more than half of their estimates. And only three of them had loss of accuracy in more than half of their estimates.

Conclusion – Reviewing the similar executed software activities, when using Planning Poker, led to more accurate software estimates in most cases, and, because of that, can improve the software development process.

Keywords: Software Metrics, Software Effort Prediction, Software Effort Estimate, Software Costs Prediction, Software Costs Estimate, Software Process Management.

LISTA DE QUADROS

QUADRO 1 – FATORES DE COMPLEXIDADE TÉCNICA.....	26
QUADRO 2 – EXEMPLO 1 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE PROJETO.....	56
QUADRO 3 – EXEMPLO 2 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE PROJETO.....	57
QUADRO 4 – EXEMPLO 3 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE PROJETO.....	59
QUADRO 5 – EXEMPLO 1 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE CARTA.....	61
QUADRO 6 – EXEMPLO 2 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE CARTA.....	61
QUADRO 7 – EXEMPLO 3 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE CARTA.....	62
QUADRO 8 – EXEMPLO 4 DE ARQUIVO YAML DO BANCO DE DADOS LEGADO PARA ENTIDADE CARTA.....	63
QUADRO 9 – EXEMPLO DE ENTRADA DE LOG DE REQUISIÇÃO GET COM SUCESSO.....	66
QUADRO 10 – EXEMPLO DE ENTRADA DE LOG DE REQUISIÇÃO POST COM SUCESSO.....	66
QUADRO 11 – EXEMPLO DE ENTRADA DE LOG DE REQUISIÇÃO POST COM SUCESSO SIMULANDO PUT.....	66
QUADRO 12 – EXEMPLO DE ENTRADA DE LOG DE REQUISIÇÃO GET CUJO RECURSO NÃO FOI ENCONTRADO.....	67
QUADRO 13 – EXEMPLO DE ENTRADA DE LOG DE REQUISIÇÃO GET COM ERROS.....	68
QUADRO 14 – VARIAÇÕES DAS CONDUÇÕES DOS EXPERIMENTOS.....	117
QUADRO 15 – LISTA DE ARTIGOS SELECIONADOS.....	147
QUADRO 16 – MELHORIAS DE ESTIMATIVA IDENTIFICADAS NOS ARTIGOS SELECIONADOS.....	160
QUADRO 17 – RESULTADOS DE EXPERIMENTOS DE IMPACTO NOS RESULTADOS DE ESTIMATIVAS BASEADAS EM JULGAMENTO DO ESPECIALISTA.....	161
QUADRO 18 – NATUREZAS DAS ATIVIDADES DISCUTIDAS ENTRE OS ARTIGOS SELECIONADOS.....	162
QUADRO 19 – HORAS DE TRABALHO ESTIMADAS POR NATUREZA.....	162
QUADRO 20 – LISTA DE ARTIGOS SELECIONADOS.....	175

LISTA DE FIGURAS

FIGURA 1 – DIAGRAMA DO FLUXO DE EXECUÇÃO DO PLANNING POKER.....	32
FIGURA 2 – DIAGRAMA DO FLUXO DE EXECUÇÃO DO PLANNING POKER COM REVISÃO DE ATIVIDADES EXECUTADAS.....	38
FIGURA 3 – DIAGRAMA DO FLUXO DE EXECUÇÃO DO EXPERIMENTO – PRIMEIRA PARTE.....	40
FIGURA 4 – DIAGRAMA DO FLUXO DE EXECUÇÃO DO EXPERIMENTO – SEGUNDA PARTE.....	41
FIGURA 5 – DIAGRAMA DO FLUXO DE EXECUÇÃO DO EXPERIMENTO – TERCEIRA PARTE.....	42
FIGURA 6 – DIAGRAMA DE CLASSES DA PRIMEIRA VERSÃO DA FERRAMENTA DE COLETA DE DADOS DOS EXPERIMENTOS.....	48
FIGURA 7 – RASCUNHO DE TELA DE TELA DE CADASTRO DE DADOS DO PARTICIPANTE.....	50
FIGURA 8 – RASCUNHO DE TELA DE ANÁLISE DE REQUISITOS E DEFINIÇÃO DE ESTIMATIVA FINAL.....	51
FIGURA 9 – RASCUNHO DE TELA DE CADASTRO DE ESTIMATIVAS INDIVIDUAIS PARA CADA RODADA DO PLANNING POKER.....	52
FIGURA 10 – RASCUNHO DE TELA DA TELA DE BUSCA NO REPOSITÓRIO DE ATIVIDADES EXECUTADAS.....	53
FIGURA 11 – TABELAS DO BANCO DE DADOS DA FERRAMENTA DE RECUPERAÇÃO DE LOGS DE REQUISIÇÃO DE NAVEGADOR.....	70
FIGURA 12 – CAPTURA DA TELA DE LISTAGEM DE GRUPOS DE ESTIMATIVA CADASTRADOS.....	72
FIGURA 13 – CAPTURA DA TELA DE CADASTRO DE GRUPO DE ESTIMATIVA.....	73
FIGURA 14 – CAPTURA DA TELA INICIAL DE SELEÇÃO E CADASTRO DE PARTICIPANTES.....	74
FIGURA 15 – CAPTURA DA TELA DE CADASTRO DE PARTICIPANTE.....	75
FIGURA 16 – CAPTURA DA TELA DE REVISÃO DO CADASTRO DE PARTICIPANTE.....	76
FIGURA 17 – CAPTURA DA TELA DE ALTERAÇÃO CADASTRO DE PARTICIPANTE.....	77
FIGURA 18 – CAPTURA DA TELA DE LISTAGEM DE REQUISITOS A ESTIMAR.....	78
FIGURA 19 – CAPTURA DA TELA DE EXIBIÇÃO DO REQUISITO A ESTIMAR E DE EXECUÇÃO DAS RODADAS DE PLANNING POKER.....	79
FIGURA 20 – CAPTURA DA TELA DE EXIBIÇÃO DO REQUISITO A ESTIMAR E DE DEFINIÇÃO DA ESTIMATIVA FINAL.....	80

FIGURA 21 – CAPTURA DA TELA DE EXIBIÇÃO DO REQUISITO ESTIMADO.....	81
FIGURA 22 – CAPTURA DA TELA DE BUSCA DE ATIVIDADES EXECUTADAS.....	82
FIGURA 23 – CAPTURA DA TELA DE BUSCA DE ATIVIDADES EXECUTADAS COM OS RESULTADOS ENCONTRADOS.....	82
FIGURA 24 – CAPTURA DA TELA DE BUSCA DE ATIVIDADES EXECUTADAS COM DETALHES DA ATIVIDADE EXECUTADA.....	83
FIGURA 25 – TABELAS DO BANCO DE DADOS DA FERRAMENTA DE COLETA.....	85
FIGURA 26 – TABELA DO BANCO DE DADOS DA FERRAMENTA DE COLETA PARA LOG INTERNO DE ACESSO PELO NAVEGADOR NA PRIMEIRA ITERAÇÃO.....	86
FIGURA 27 – TABELA DO BANCO DE DADOS DA FERRAMENTA DE COLETA PARA LOG INTERNO DE ACESSO PELO NAVEGADOR NA SEGUNDA ITERAÇÃO.....	86
FIGURA 28 – CAPTURA DA TELA APRIMORADA DE EXIBIÇÃO DO REQUISITO ESTIMADO.....	88
FIGURA 29 – CAPTURA DA TELA APRIMORADA DE BUSCA DE ATIVIDADES EXECUTADAS.....	90
FIGURA 30 – CAPTURA DA TELA APRIMORADA DE BUSCA DE ATIVIDADES EXECUTADAS.....	92
FIGURA 31 – CAPTURA DA TELA COM A TABELA DE COMPARAÇÃO DO RESULTADO DO GRUPO DE ESTIMADORES.....	93
FIGURA 32 – CAPTURA DA TELA DE ANÁLISE DE MELHORIA DE ACURÁCIA DO GRUPO DE ESTIMADORES ENTRE ETAPAS COM OS DADOS DE APRESENTAÇÃO DO GRUPO DE ESTIMADORES SELECIONADO.....	95
FIGURA 33 – CAPTURA DA TELA DA LISTA DE GRUPOS DE ESTIMADORES A SELECIONAR PARA ANÁLISE DE RESULTADOS.....	96
FIGURA 34 – CAPTURA PARCIAL DE TELA COM A TABELA DE ANÁLISE DE MELHORIA DE ACURÁCIA DO GRUPO DE ESTIMADORES ENTRE ETAPAS COM OS RESULTADOS DE UMA ETAPA ISOLADA.....	97
FIGURA 35 – CAPTURA PARCIAL DE TELA COM A TABELA DE ANÁLISE DE MELHORIA DE ACURÁCIA DO GRUPO DE ESTIMADORES ENTRE ETAPAS COM A COMPARAÇÃO DOS RESULTADOS DE CADA ETAPA.....	99
FIGURA 36 – CAPTURA PARCIAL DE TELA COM A TABELA DE COMPARAÇÃO DE MELHORIA DE ACURÁCIA DO GRUPO DE ESTIMADORES ENTRE GRUPOS DE ESTIMADORES.....	102
FIGURA 37 – CAPTURA DA TELA DE LINKS DE COMPARAÇÕES DE RESULTADO POR REQUISITO.....	103

FIGURA 38 – CAPTURA DA TELA DE LISTA DE TIMES QUE RESPONDERAM CADA REQUISITO.....	104
FIGURA 39 – CAPTURA DA TELA DE LINKS DE CONTAGEM DE RESULTADO POR PARTICIPANTE.....	109
FIGURA 40 – CAPTURA PARCIAL DE TELA DE ANÁLISE DE MELHORIA DE ACURÁCIA POR ESTIMADOR PARTICIPANTE INDIVIDUAL EM CADA GRUPO.....	110
FIGURA 41 – CAPTURA PARCIAL DE TELA DE ANÁLISE DE MELHORIA DE ACURÁCIA POR ESTIMADOR PARTICIPANTE INDIVIDUAL INDEPENDENTE DO GRUPO DE ESTIMADORES.....	112
FIGURA 42 – CAPTURA DA TELA DE LINKS DE DADOS DE CORRELAÇÃO.....	113
FIGURA 43 – CAPTURA DA TABELA DE RESULTADOS DE TESTES DE CORRELAÇÃO DE DADOS DOS EXPERIMENTOS.....	115
FIGURA 44 – CAPTURA DO QUADRO DE LEGENDA DE SIGLAS USADAS NA EXIBIÇÃO DE RESULTADOS DE TESTES DE CORRELAÇÃO DE DADOS DOS EXPERIMENTOS.....	116
FIGURA 45 – CAPTURA DA TABELA DE RESULTADOS DE TESTES DE CORRELAÇÃO DE DADOS DOS EXPERIMENTOS PARA OS GRUPOS SELECIONADOS.....	124
FIGURA 46 – GRÁFICO DE DISPERSÃO ENTRE A MÉDIA DAS NOTAS DE CADA GRUPO E O AUMENTO RELATIVO DA MÉDIA DA DIFERENÇA ABSOLUTA.....	125
FIGURA 47 – MAIOR REDE DE COOPERAÇÃO ENTRE AUTORES.	154
FIGURA 48 – REDE DE COOPERAÇÃO ENTRE AUTORES DE AUTO FATOR DE PUBLICAÇÃO.....	155
FIGURA 49 – REDE DE COOPERAÇÃO INFLUENCIADA PRINCIPALMENTE POR MOATAZ AHMED.....	155
FIGURA 50 – REDE DE COOPERAÇÃO INFLUENCIADO PRINCIPALMENTE POR LEFTERIS ANGELIS.....	156
FIGURA 51 – REDE DE COOPERAÇÃO INFLUENCIADO PRINCIPALMENTE POR ALEŠ ŽIVKOVIČ.....	156
FIGURA 52 – DIAGRAMA DO FLUXO DE EXECUÇÃO DO PLANNING POKER.....	178

LISTA DE TABELAS

TABELA 1 – TABELA DE ANÁLISE DE RESULTADOS POR REQUISITO PARA UMA ETAPA, PARTE 1.....	105
TABELA 2 – TABELA DE ANÁLISE DE RESULTADOS POR REQUISITO PARA UMA ETAPA, PARTE 2.....	106
TABELA 3 – TABELA DE ANÁLISE DE RESULTADOS POR REQUISITO PARA UMA ETAPA, PARTE 3.....	107
TABELA 4 – TABELA DE CONTAGEM DE RESULTADOS DOS GRUPOS DE ESTIMADORES POR REQUISITO.....	108
TABELA 5 – COMPARAÇÃO INICIAL DOS TIMES EM SEUS EXPERIMENTOS.....	120
TABELA 6 – COMPARAÇÃO DA MELHORIA DE ACURÁCIA DOS TIMES EM SEUS EXPERIMENTOS.....	121
TABELA 7 – COMPARAÇÃO DA MELHORIA DE ACURÁCIA DOS GRUPOS ENTRE AS ETAPAS.....	122
TABELA 8 – RESPOSTAS DO QUESTIONÁRIO DE USABILIDADE..	127
TABELA 9 – RESPOSTAS DO QUESTIONÁRIO DE SATISFATIBILIDADE.....	127
TABELA 10 – REVISTAS SELECIONADAS.....	143
TABELA 11 – NÚMERO DE ARTIGOS AVALIADOS A CADA ESTÁGIO DO PROCESSO DE SELEÇÃO.....	144
TABELA 12 – NÚMERO DE ARTIGOS POR AUTOR ENTRE OS SELECIONADOS.....	152
TABELA 13 – NÚMERO DE ARTIGOS SELECIONADOS POR REVISTA.....	157
TABELA 14 – CONTAGEM E COMPARAÇÃO DE ABORDAGENS ENCONTRADAS E COMPARAÇÃO COM OS RESULTADOS ENCONTRADOS POR JØRGENSEN E SHEPPERD (2007).....	158
TABELA 15 – HORAS DE TRABALHO ESTIMADAS POR NATUREZA.....	173

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS.....	16
1.2 ORGANIZAÇÃO DO DOCUMENTO.....	17
2 ESTIMATIVA DE ESFORÇO DE SOFTWARE.....	18
2.1 LEVANTAMENTO E ANÁLISE DE REQUISITOS.....	18
2.2 ABORDAGENS DE ESTIMATIVAS.....	18
2.2.1 Analogia.....	19
2.2.2 Árvores de Classificação e Regressão.....	20
2.2.3 Combinação de Estimativas.....	20
2.2.4 Julgamento de Especialista.....	21
2.2.5 Pontos de Função.....	24
2.2.6 Rede Bayesiana.....	27
2.2.7 Redes Neurais Artificiais.....	27
2.2.8 Regressão.....	28
2.2.9 Outros.....	30
2.3 PLANNING POKER.....	31
2.4 EDUCAÇÃO E TREINAMENTO EM ANÁLISE DE REQUISITOS.....	36
3 REVISÃO DE ATIVIDADES EXECUTADAS.....	37
4 MATERIAIS E MÉTODOS.....	39
4.1 TERMOS UTILIZADOS.....	44
4.2 TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	44
4.3 QUESTIONÁRIO DE AVALIAÇÃO DA FERRAMENTA.....	44
4.4 CARTAS POSSÍVEIS.....	45
4.5 ANÁLISE ESTATÍSTICA.....	45
4.6 MODELAGEM DA FERRAMENTA DE COLETA DE DADOS DO EXPERIMENTO.....	46
4.6.1 Banco de Dados Legado.....	46
4.6.2 Diagrama de Classes.....	46
4.6.3 Rascunhos de Tela.....	49
4.6.3.1 Cadastro de Dados do Participante.....	49
4.6.3.2 Análise de Requisitos e Definição de Estimativa Final.....	50
4.6.3.3 Cadastro de Estimativas Individuais por Rodada.....	51
4.6.3.4 Busca no Repositório de Atividades Executadas.....	52
5 FERRAMENTAS DESENVOLVIDAS.....	54
5.1 FERRAMENTA DE MIGRAÇÃO DE DADOS LEGADOS PARA A FERRAMENTA DE COLETA DE DADOS.....	54
5.1.1 Entidade Projeto.....	54
5.1.2 Entidade Carta.....	60
5.1.3 Seleção das Atividades e Projetos Migrados.....	64
5.2 FERRAMENTA DE MIGRAÇÃO DE NOTAS DOS PARTICIPANTES.....	65
5.3 FERRAMENTA DE RECUPERAÇÃO DE REQUISIÇÕES DO NAVEGADOR.....	65
5.3.1 Exemplos de Entradas de Log.....	65
5.3.2 Banco de Dados.....	68
5.4 FERRAMENTA DE COLETA DE DADOS.....	71
5.4.1 Telas.....	71
5.4.1.1 Definição do Grupo de Estimativa.....	71
5.4.1.2 Seleção e Cadastro dos Participantes.....	73
5.4.1.3 Definição de Estimativas.....	77

5.4.1.4 Repositório de Atividades Executadas.....	81
5.4.2 Registro de Eventos.....	83
5.4.3 Banco de Dados.....	84
5.4.4 Alterações Após Primeira Iteração.....	86
5.4.4.1 Adicionar Validação para Evitar que os Usuários Insiram Estimativas sem Realizar as Rodadas.....	87
5.4.4.2 Separar Dados de Atividade e Dados de Projeto.....	87
5.4.4.3 Apenas Exibir o <i>Link</i> “definir valor final” se já Houver ao menos uma Rodada Executada.....	88
5.4.4.4 Mudar as <i>Labels</i> de Valores Relativos ao Momento e ao Conteúdo Entregue no Projeto para Ficarem Mais Claros.....	89
5.4.4.5 Colocar o <i>Link</i> dos Dados Históricos Após a Seleção do Grupo de Estimadores para Manter <i>Link</i> com o Grupo.....	89
5.4.4.6 Não Exibir na Busca os Projetos sem Requisitos da Busca.....	89
5.4.4.7 Na Busca por Palavra Chave, Buscar Também em Tecnologias e Objetivos dos Projetos.....	90
5.4.4.8 Alterar <i>Label</i> na Busca por “Palavras-chave”.....	90
5.4.4.9 Na Busca por Palavra Chave, Usar as Várias Palavras como Palavras-Chave Separadas e Ordenar os Resultados Contando as Palavras-Chave Encontradas....	90
5.4.4.10 Adicionar Número de Resultados Encontrados na Busca.....	91
5.4.5 Apresentação do Resultado para os Participantes.....	92
5.5 FERRAMENTA DE ANÁLISE DE DADOS.....	94
5.5.1 Análise de Melhoria do Grupo de Estimadores.....	94
5.5.1.1 Análise de Melhoria de Acurácia do Grupo de Estimadores Entre Etapas.....	94
5.5.1.2 Comparação da Melhoria de Acurácia entre os Grupos de Estimadores.....	101
5.5.2 Análise de Melhoria por Requisitos.....	103
5.5.2.1 Lista de Times que Responderam cada Requisito.....	104
5.5.2.2 Análise de Resultados por Requisito.....	105
5.5.2.3 Contagem de Resultados por Requisito.....	107
5.5.3 Análise de Melhoria por Participante.....	108
5.5.3.1 Contagem de Melhoria por Participante Nos Times.....	109
5.5.3.2 Contagem de Melhoria por Participante Independentes dos Times.....	111
5.5.4 Análise de Correlação.....	113
5.5.4.1 Dados de Correlação para Todos Experimentos.....	113
5.5.4.2 Dados de Correlação para Um Experimento Específico.....	116
6 EXPERIMENTO.....	117
6.1 DADOS COLETADOS NO EXPERIMENTO.....	119
6.2 ANÁLISE DOS RESULTADOS.....	122
7 AVALIAÇÃO DE USABILIDADE DA FERRAMENTA.....	126
7.1 QUESTIONÁRIO.....	126
7.2 RESULTADOS DO QUESTIONÁRIO DE AVALIAÇÃO DA FERRAMENTA DE COLETA DE DADOS.....	126
7.3 ANÁLISE DA USABILIDADE.....	128
8 CONCLUSÃO.....	129
A REVISÕES SISTEMÁTICAS DA LITERATURA.....	138
A.1 REVISÃO SISTEMÁTICA DA LITERATURA EM ESTIMATIVA DE ESFORÇO DE SOFTWARE.....	138
A.1.1 Protocolo de Pesquisa de Artigos da Literatura.....	138
A.1.1.1 Questões de Pesquisa de Artigos da Literatura.....	139
A.1.1.1.1 Pesquisadores Influentes.....	139

A.1.1.1.2 Grupos de Pesquisa Influentes.....	139
A.1.1.1.3 Revistas Influentes.....	139
A.1.1.1.4 Principais Abordagens para as Estimativas de Esforço de Software.....	140
A.1.1.1.5 Melhoras Significativas Encontradas em Pesquisas.....	140
A.1.1.1.6 Tipos de Atividades de Software cuja Acurácia da Estimativa de Esforço é Menor.....	140
A.1.1.2 Estratégia de Pesquisa de Artigos da Literatura.....	141
A.1.1.2.1 Seleção de Bases de Dados.....	141
A.1.1.2.2 Seleção de Revistas.....	142
A.1.1.3 Realização da Pesquisa de Artigos da Literatura.....	143
A.1.1.3.1 Seleção de Artigos para Inclusão.....	143
A.1.1.3.2 Verificação da Seleção.....	144
A.1.1.3.3 Estratégia de Extração de Dados.....	144
A.1.1.3.4 Desvios do Protocolo.....	145
A.1.1.3.5 Artigos Selecionados.....	146
A.1.2 Resultados e Discussão da Revisão Sistemática da Literatura em Estimativa de Esforço de Software.....	151
A.1.2.1 QP1. Quem São os Principais Pesquisadores da Área de Estimativas de Esforço de Desenvolvimento de Software?.....	151
A.1.2.2 QP2. Quais São os Principais Grupos de Pesquisa?.....	152
A.1.2.3 QP3. Que Revistas Costumam Publicar na Área?.....	157
A.1.2.4 QP4. Quais São as Principais Abordagens para Estimativa de Esforço de Software?.....	157
A.1.2.5 QP5. Que Estudos Mostraram Melhoras Significativas na Predição de Tempo de Execução das Atividades de Software? E para Quais Abordagens?.....	159
A.1.2.6 QP6. Quais os Tipos de Atividades de Software que Divergem Mais Entre o Tempo Estimado e o Tempo Realizado?.....	161
A.1.3 Discussão da Revisão Sistemática da Literatura em Estimativa de Esforço de Software.....	163
A.1.4 Conclusão da Revisão Sistemática da Literatura em Estimativa de Esforço de Software.....	170
A.2 REVISÃO DO ESTADO DA ARTE EM <i>PLANNING POKER</i>	172
A.2.1 Protocolo de Pesquisa de Artigos da Literatura.....	172
A.2.1.1 Estratégia de Pesquisa de Artigos da Literatura.....	172
A.2.1.1.1 Seleção de Bases de Dados.....	172
A.2.1.1.2 Realização da Pesquisa de Artigos da Literatura.....	173
A.2.1.1.2.1 Seleção de Artigos para Inclusão.....	173
A.2.1.1.2.2 Verificação da Seleção.....	173
A.2.1.1.2.3 Desvios do Protocolo.....	174
A.2.1.1.2.4 Artigos Selecionados.....	174
A.2.2 Introdução a <i>Planning Poker</i>	177
A.2.3 Discussão da Revisão do Estado da Arte em <i>Planning Poker</i>	179
A.2.4 Conclusão da Revisão do Estado da Arte em <i>Planning Poker</i>	181
B TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	183
C QUESTIONÁRIO DE AVALIAÇÃO DA FERRAMENTA.....	184

1 INTRODUÇÃO

A estimativa de esforço de desenvolvimento de software é o processo de prever o esforço necessário para desenvolver um software. Entregar o software no prazo é uma preocupação de muitas organizações. Subestimar o esforço para o desenvolvimento de softwares pode levar à entrega dos mesmos com uma qualidade menor, além de causar impacto na reputação e na competitividade da organização. Porém, superestimar o custo pode resultar em perda de oportunidades interessantes de negócio, ou seja, uma organização pode deixar de desenvolver um novo software por considerar não ter tempo hábil ou pessoas disponíveis para isto.

Os projetos de software, em média, atingem 30% de precisão em suas estimativas de software (Jørgensen e Grimstad 2008), o que sugere que o planejamento, como é conduzido hoje, não se mostra confiável o suficiente e, desta forma, o mesmo pode levar ao fracasso do projeto.

O acompanhamento das execuções das atividades de desenvolvimento de software é crucial para a entrega do software no prazo, já que o atraso em uma atividade, se não for compensado dentro do cronograma, tende a resultar em atraso na entrada do sistema em produção (utilização pelos usuários), além de evitar subutilizar os recursos empregados no desenvolvimento. Atrasos na realização de atividades ocorrem corriqueiramente pela má distribuição de atividades entre os recursos disponíveis e pela realização de estimativas irrealistas.

A acurácia das estimativas de esforço de software depende da habilidade do profissional estimador e da sua capacidade de estimar a produtividade da sua equipe. E, como qualquer ser humano, este tem sua capacidade de decisão afetada por características ambientais ou psicológicas. Entre estes fatores, a falta de preparação para a realização da atividade de estimação faz com que o estimador não exercite o conjunto de habilidades necessárias para a realização adequada da estimativa de esforço. E, ao verificar a literatura, não foi possível encontrar nenhum dado sobre a influência da revisão de atividades executadas antes da realização de estimativas baseadas em julgamento do especialista.

Há diversos modelos de estimativa de esforço de software como um todo como, por exemplo, Pontos de Casos de Uso (Yavari et al. 2011), Pontos de Função (Robiolo e Orosco 2007) e Regressão Linear (Robiolo e Orosco 2007). Porém,

pouco se discute sobre a forma como se estimam as horas de realização de atividades individuais de desenvolvimento de software. Mesmo utilizando técnicas como *Planning Poker* (Tamrakar e Jørgensen 2012), popularizada pelo movimento de implantação de técnicas ágeis de desenvolvimento, a estimativa de tempo de execução ainda é baseada no julgamento de um especialista, onde o responsável pela estimativa tem de conhecer profundamente o objetivo da atividade e da eficiência dos profissionais que irão executá-la. Como o seu resultado é um valor estático, a técnica não é capaz de responder a mudanças de cronograma ou alocação de recursos.

O *Planning Poker* é considerado uma técnica de estimativa *top-down* baseado em julgamento de especialista, utilizada em métodos de desenvolvimento ágeis nos estágios iniciais do desenvolvimento, quando se estima o tamanho do software, sendo que o esforço calculado é baseado na velocidade do time de desenvolvimento (Dagnino 2013).

Os métodos mais comuns na indústria para realização de estimativas são os baseados em julgamento de especialista, e sendo em média tão precisos quanto o uso de modelos formais de estimativa (Jørgensen 2013b). A habilidade humana de refletir sobre suas estimativas e melhorá-las é surpreendentemente pobre (Jørgensen e Gruschke 2009). Na maioria dos casos (83%), a combinação de técnicas de estimativa é melhor que qualquer técnica isolada (Kocaguneli et al. 2012a).

As informações geradas pela discussão no *Planning Poker* não são guardadas pelos participantes, nem quando realizadas por equipes locais, pela informalidade aparente do método, nem quando realizadas por times geograficamente distribuídos, pela falta de suporte das ferramentas de mercado especializadas em *Planning Poker*. E como este conhecimento se perde, não há como aproveitá-lo em estimativas futuras.

Neste contexto, este trabalho teve como objetivo verificar a melhoria da acurácia de estimativa de esforço de desenvolvimento de software utilizando a técnica de estimativa *Planning Poker*, por meio de experimentos com e sem revisão, ou releitura, de estimativas similares anteriores. E se faz interessante por verificar um fator não abordado pelos artigos encontrados.

Com a realização dos experimentos, os resultados mostraram melhora na acurácia da estimativa, sendo recomendada a realização destas revisões em

contextos similares, melhorando assim o resultado do processo de estimativa.

Uma ferramenta web para a coleta de dados de estimativa e captura de eventos do navegador, para traçar o raciocínio dos participantes no processo de definição da estimativa, foi desenvolvida com base no processo de *Planning Poker* e amadurecida com a opinião dada pelos participantes do experimento.

A maior contribuição deste trabalho foi a criação de um método que visou estender o *Planning Poker*, corrigindo duas desvantagens deste. A primeira desvantagem é a perda completa das informações da discussão a respeito do tempo e da complexidade de implementação dos requisitos, que ficam apenas na memória dos participantes, corrigida pela ferramenta com o armazenamento das argumentações principais utilizadas nas discussões. Já a segunda desvantagem está relacionada à discussão ser apenas local, em vez de utilizar o legado relativo às estimativas anteriores, o que prejudica o processo de melhoria das estimativas de software. A ferramenta possibilita o uso desse legado.

1.1 OBJETIVO GERAL E OBJETIVOS ESPECÍFICOS

O objetivo geral deste trabalho é verificar o impacto na acurácia da realização de estimativas de esforço de desenvolvimento de software, utilizando *Planning Poker* e a revisão de estimativas anteriores.

Os objetivos específicos da realização dos experimentos, funcionando como subprodutos da busca pelo objetivo principal, são:

- Investigar métricas de desenvolvimento de software para verificar o impacto da acurácia.
- Desenvolver uma ferramenta web para coleta dos dados gerados durante a execução do *Planning Poker*.
- Desenvolver uma ferramenta web para apresentação de relatórios de desempenho de estimativa, com taxa de acurácia e suas correlações.
- Investigar possíveis correlações entre o rendimento acadêmico dos participantes da pesquisa e a melhoria da acurácia de estimativa.
- Realizar e analisar resultados da avaliação de usabilidade da ferramenta de coleta de dados.

1.2 ORGANIZAÇÃO DO DOCUMENTO

Este documento está estruturado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica, o Capítulo 3 descreve a metodologia adotada, o Capítulo 4 apresenta a proposta do trabalho, o Capítulo 5 descreve as ferramentas desenvolvidas para a completa realização dos experimentos, o Capítulo 6 descreve os experimentos realizados e seus resultados, o Capítulo 7 apresenta o estudo de usabilidade da ferramenta utilizada pelos participantes; finalmente, o Capítulo 8 apresenta as conclusões obtidas durante a realização dos experimentos e análise dos resultados.

O Apêndice deste documento está estruturado da seguinte forma: o Capítulo A apresenta a revisão da arte em estimativa de software e *Planning Poker*; o Capítulo B apresenta o termo de consentimento livre e esclarecido; o Capítulo C apresenta o questionário de usabilidade, no formato em que foi apresentado aos participantes.

2 ESTIMATIVA DE ESFORÇO DE SOFTWARE

Com o objetivo de dar base a discussões e descrever o cenário de pesquisa atual, foram realizadas revisões da literatura. A pesquisa para compreensão do estado da arte foi realizada em duas etapas. A primeira etapa foi realizada no formato de revisão sistemática da literatura examinando o estado da arte da área de pesquisa de Estimativa de Esforço de Software e reportando as conclusões cabíveis. Já na segunda etapa, foi realizada uma revisão do estado da arte específica para o contexto de *Planning Poker*. Estas revisões sistemáticas são detalhadas no Apêndice A.

2.1 LEVANTAMENTO E ANÁLISE DE REQUISITOS

Antes de desenvolver um sistema, é necessário compreender o que este deve fazer e como o uso deste pode auxiliar indivíduos ou empresas a atingirem seus objetivos. O que envolve, segundo Sommerville (2005), compreender o domínio da aplicação, restrições do sistema operacional, funcionalidades exigidas pelos envolvidos no projeto, e características como desempenho, segurança e confiabilidade. E como para estimar o esforço de desenvolvimento de software é necessário entender o que este deve realizar, a acurácia das estimativas está diretamente ligada a qualidade dos produtos da engenharia de requisitos.

Segundo Castro-Herrera e Cleland-Huang (2010), esta é uma fase crítica do projeto, e diversas pesquisas e estudos de caso têm mostrado que requisitos incompletos ou incorretos é uma das causas principais da falha do projeto e podem levar à perda de milhões de dólares todo ano.

2.2 ABORDAGENS DE ESTIMATIVAS

Várias abordagens para estimativa de esforço de software foram propostas na literatura. Jørgensen e Shepperd (2007) dividem estas abordagens em 12

categorias: “Analogia”, “Árvores de Classificação e Regressão”, “Combinação de Estimativas”, “Julgamento de Especialista”, “Pontos de Função”, “Rede Bayesiana”, “Redes Neurais Artificiais”, “Regressão”, “Simulação”, “Teoria”, “*Work Break-Down*”, e “Outros”.

Segundo Gruschke e Jørgensen (2008), estimativas baseadas em Julgamento do Especialista tiveram diferença entre esforço estimado mediano e esforço real de 8% a 63%, o que aponta para a instabilidade da técnica. Porém, mesmo a melhor das técnicas baseadas em aprendizado de máquina apresentou diferença entre esforço estimado mediano e esforço real de 27,5% a 60,8%, o que também se mostra insatisfatório (Dejaeger et al 2012). Na maioria dos casos, a combinação de técnicas de estimativa é melhor que qualquer técnica isolada (Kocaguneli et al 2012).

Nas próximas seções, as abordagens de estimativa de software são apresentadas segundo a categorização de Jørgensen e Shepperd (2007).

2.2.1 Analogia

Analogia, ou *Case-based Reasoning* (CBR), é um ramo da Inteligência Artificial (Mendes e Mosley 2008) e trabalha de modo similar a como o especialista estima esforço de software (Dejaeger et al 2012) (Mittas e Angelis 2013), provendo estimativas de esforço para novos projetos ao comparar as características do projeto analisado com projetos cujo esforço é conhecido (Mendes e Mosley 2008) (Kocaguneli et al. 2012b). Essa comparação geralmente utiliza distância euclidiana e características redimensionadas para encontrar os casos mais similares (Dejaeger et al. 2012) (Kocaguneli et al. 2012b).

Na analogia podem ser usadas características como tamanho de software, esforço, e duração do projeto (Kocaguneli et al. 2012b). A estimativa de esforço resultante dessa abordagem é determinada pela média do esforço dos casos similares encontrados (Dejaeger et al. 2012).

No entanto, não há consenso com relação a quais combinações de parâmetros são melhores para prover as melhores estimativas por analogia, de modo que a escolha dos parâmetros deve se basear nos dados disponíveis no

banco de dados histórico (Mendes e Mosley 2008). Além disso, não há métodos simples para identificar projetos cujas características sejam anormais (Keung et al. 2008) e é necessário o uso de ferramentas específicas para aplicação de métodos baseados em Analogia (Kitchenham et al. 2007).

2.2.2 Árvores de Classificação e Regressão

A classificação e árvore de regressão procura os casos mais parecidos no conjunto de dados para determinar o esforço necessário para um novo projeto (Dejaeger et al. 2012) (Kocaguneli et al. 2012b). O número de casos mais parecidos, ou vizinhos mais próximos, a serem levados em consideração, dos quais a média seria utilizada, tem sido discutido pela comunidade de pesquisa (Kocaguneli et al. 2012b).

O algoritmo Classification and Regression Tree, ou CART, constrói uma árvore binária separando recursivamente o conjunto de dados até o critério de parada ser satisfeito (Dejaeger et al. 2012) (Mittas e Angelis 2013). No caso de Kocaguneli et al. (2012b), o critério de parada definido foi quando a variância da árvore inferior é mais alta que a variância da árvore superior. A facilidade de compreensão das árvores de regressão pode ser considerada um dos pontos fortes dessa abordagem (Dejaeger et al. 2012). Porém, nos testes realizados por Mittas e Angelis (2013), CART obteve a pior acurácia para estimativa de esforço de software.

2.2.3 Combinação de Estimativas

Na combinação de métodos de estimativas são construídos modelos, para diferentes métodos de aprendizagem de máquina, cujas estimativas são combinadas (Kocaguneli et al. 2012a).

Combinações de métodos são úteis por combinar características de funcionamento, permitindo que um método melhore a acurácia do outro (Kocaguneli et al. 2012a).

Kocaguneli et al. (2012a) combinaram a execução dos métodos de

aprendizagem de máquina entre si:

- Árvore de Classificação e Regressão com poda.
- Árvore de Classificação e Regressão sem poda.
- Rede Neural com duas camadas ocultas.
- Regressão Linear Simples.
- Regressão de Componentes Principais.
- Regressão por Mínimos Quadrados Parciais.
- Regressão *Stepwise*.
- Estimativa de Esforço Baseada em Analogia, levando em conta apenas o vizinho mais próximo.
- Estimativa de Esforço Baseada em Analogia, levando em conta apenas os cinco vizinhos mais próximos.

O resultado da comparação, realizada por Kocaguneli et al. (2012a), foi que os resultados destas combinações foi melhor que qualquer um dos métodos isolados. Além disto, a média entre resultados de estimativas isoladas geralmente obtém um resultado mais acurado que os resultados de estimativas isoladas.

2.2.4 Julgamento de Especialista

Julgamento do Especialista é baseado na realização de *brainstorming*¹ de um ou mais especialistas que têm experiência em projetos similares, chegando a um consenso que produz uma estimativa (Buglione e Ebert 2011).

Estimativa de esforço baseada em julgamento do especialista é a técnica mais frequentemente utilizada na indústria de software, e é em média pelo menos tão acurada quanto modelos formais de estimativa (Jørgensen 2013b).

O uso de processos inconscientes de julgamento é essencial na maioria das estimativas de esforço de desenvolvimento de software, como parte de estimativas baseadas em julgamento do especialista ou na entrada de dados em modelos de estimativa (Jørgensen e Grimstad 2011).

Jørgensen e Grimstad (2011) defendem que estes processos inconscientes

¹ Técnica de discussão em grupo que se vale da contribuição espontânea de ideias por parte de todos os participantes, no intuito de resolver algum problema ou de conceber um trabalho criativo.

de julgamento podem ser afetados por informações irrelevantes como, por exemplo, a expectativa do cliente ao chamar a tarefa de “desenvolvimento de uma nova funcionalidade” ou “uma extensão”, o excesso de informações, e informações que induzem otimismo como “oportunidades futuras”.

Jørgensen (2009) apresenta várias recomendações para o profissional que avalia os licitantes das empresas, com o objetivo de evitar decidir por empresas cujo orçamento seja superotimista. E assim como em “Guidelines for Software Development Effort Estimation” (Basten e Sunyaev 2011), o autor apenas descreve alguns fatores que podem levar a decidir pela licitante não competente ou cujo orçamento superotimista levará a má qualidade de software, mas sem apontar os problemas mais comuns ou o seu impacto no processo de avaliação de orçamento.

Jørgensen e Grimstad (2008) testaram a hipótese de que o impacto de informações irrelevantes afeta significativamente a acurácia da estimativa de software em uma pesquisa com 170 profissionais, distribuídos aleatoriamente entre 4 grupos. Os três primeiros grupos receberam uma grande quantidade de informações irrelevantes, enquanto que o quarto grupo apenas recebeu informações relevantes para a realização da estimativa. No segundo grupo, os profissionais foram orientados a destacar as informações relevantes. No terceiro grupo, os profissionais foram orientados a riscar as informações irrelevantes.

Jørgensen e Grimstad (2008) observaram que destacar informações relevantes ou riscar informações irrelevantes não surtiu o efeito desejado, além de que a presença das informações irrelevantes causou aumento de 40% a 60% em homens-horas estimados. Porém, os autores não levaram em consideração aspectos pessoais dos participantes que poderiam ter enriquecido a pesquisa e levantado outras questões, como sexo, experiência, ou aspectos culturais.

Jørgensen e Grimstad (2011) retomaram a pesquisa de 2008 (Jørgensen e Grimstad 2008), conduzindo experimentos com documentos de análise com diferentes quantidades de informações irrelevantes em ambientes de laboratório (acadêmico) e de campo (indústria). Outros vieses causados por informações irrelevantes foram levantados e apresentados por Jørgensen (2013b) em uma pesquisa sobre a influência que estimadores sofrem ao estimar mais de um projeto ou atividade, por imaginar relações entre projetos e/ou atividades independentes de forma inconsciente.

Embora a estimativa seja uma atividade que exija conhecimento e

familiaridade com o ambiente de produção, há sempre vieses humanos. Por exemplo, uma atividade, apenas tendo sua classificação alterada de “pequena alteração” para “nova funcionalidade”, geraria a expectativa de demandar mais esforço (Jørgensen 2013b). E para diminuir a influência dos vieses humanos, entre outras considerações, o autor indica a combinação de várias estimativas, sendo realizadas por mais de um estimador e por mais de um método.

Jørgensen et al. (2009) apresentam uma discussão entre os benefícios do uso de métodos de estimativa baseados em Julgamento do Especialista e benefícios do uso de métodos baseados em modelo. Em seu artigo são levantados os problemas de cada um, como a necessidade de manter dados históricos íntegros para alimentar os modelos, a falta de visão de negócio destes modelos, a dificuldade humana de lidar com seus vieses individuais e sua dificuldade de realizar estimativas objetivas ou de aprender com os erros passados. Aprendizado este que Jørgensen e Gruschke (2009) afirmam que pode, sim, diminuir a influência dos fatores negativos na estimativa, sugerindo a revisão de todas as estimativas, forçando o estimador a refletir sobre suas estimativas antigas e aprender com os erros.

Hearty et al. (2009) apresentam a aplicação de técnica de Redes Bayesianas para predição de velocidade da execução de iterações para projetos realizados seguindo a metodologia de desenvolvimento ágil, *Extreme Programming*. Porém, como não foram realizados testes com conjuntos de dados públicos, não há como saber se haveriam métodos com melhor desempenho para o caso abordado, como por exemplo, métodos baseados em Regressão, por serem mais apropriados para problemas desta natureza (Mendes e Mosley 2008).

Basten e Sunyaev (2011), assim como Jørgensen (2013b), apresentam uma lista de fatores que podem causar vieses na realização de estimativas de software. Estes fatores foram encontrados ao realizar a análise de 50 pesquisas empíricas publicadas desde 1990, identificando fatores que tendem a afetar a acurácia da estimativa de software. Agrupando os fatores por “Processo”, “Estimador”, “Projeto”, e “Contexto”. Basten e Sunyaev (2011), neste artigo, procuram diminuir o impacto dos vieses presentes no processo de estimativa de software, porém não apontam quais fatores impactam mais ou quais são os mais comuns, nem listam os artigos nos quais estes fatores foram encontrados.

Jørgensen e Grimstad (2012), realizaram uma pesquisa com 374

desenvolvedores de 6 países, conduzindo 3 experimentos com objetivo de relacionar problemas e acurácia de estimativa e os atributos dos desenvolvedores estilo de pensamento, como se veem, nacionalidade, experiência, habilidade, educação, sexo e papel organizacional. Porém, a pesquisa teve conclusão contraditória para a relação dos atributos de experiência, de educação e de papel organizacional e, embora afirmem que vieses podem depender da cultura ou de outros fatores contextuais, não indicam quais fatores ou quanto estes influenciam.

Nan e Harter (2009), diferente de Jørgensen e Grimstad (2012), focam na pressão exercida pelo cronograma e pelo orçamento. Tendo como base a definição de pressão como a relação entre “tempo/orçamento estimado” e “tempo/orçamento negociado”, da análise de 14 artigos de 1974 a 2005, e de um conjunto de dados privado de uma empresa de tecnologia de 25 bilhões de dólares de receita anual, foi analisada a relação encontrada entre cronograma e pressão e orçamento e pressão e se mostraram não lineares e benéficos até 10% de pressão, porém acima deste valor se mostraram prejudiciais. Pelos testes serem de simples implementação e utilizarem apenas um conjunto de dados privado, se torna interessante realizar testes em conjuntos de dados públicos para, além de verificar as conclusões dadas, cruzar com as técnicas de estimativa de software para verificar a relação entre as duas.

2.2.5 Pontos de Função

O método de Pontos de Função foi originalmente introduzido por Albrecht e Gaffney (1983) para mensurar o tamanho de sistemas de processamentos de dados, do ponto de vista do usuário, para estimar o esforço de desenvolvimento.

A ideia básica da Análise de Pontos de Função² (FPA) é que as funcionalidades entregues ao usuário podem ser avaliadas ao levar em conta os dados utilizados na aplicação para prover as funções requisitadas, e as transações (entradas e saídas da aplicação). Ambos os dados e transações são avaliadas no ponto de vista do usuário, sendo contados a partir da especificação de requisitos (Lavazza et al. 2013).

² Em inglês, *Function Points Analysis*.

Em FPA, os requisitos funcionais do usuário são modelados como um conjunto de Componentes Funcionais Básicos (BFC), em que cada BFC é medido. O tamanho da aplicação é obtido pela soma dos tamanhos dos BFCs. BFCs são funções de dados³ (DF), que são classificados em arquivos lógicos internos (ILF) e arquivos de interface externa (EIF), e funções de transacionais, que são classificados e em entradas externas (EI), saídas externas (EO), e consultas externas (EQ) dependendo da intenção principal do processo. Cada função, seja de dados ou transacional, contribui com o número de pontos de função, de acordo com a tabela de referência por tipo e complexidade, que somados geram o valor de Pontos de Função Não-Ajustados (UFP) (Lavazza et al. 2013), como na fórmula (1).

$$UFP = \sum EI + \sum EO + \sum EQ + \sum ILF + \sum EIF \quad (1)$$

Como o UFP calcula o tamanho do software, sem levar em consideração aspectos de ambiente ou requisitos não funcionais, este valor deve ser ajustado com o Fator de Complexidade Técnica (TCF), como na fórmula (2).

$$FP = UFP \times TCF \quad (2)$$

Para calcular o TCF, são somados os impactos para os 14 fatores descritos no Quadro 1 (Symons 1988), e calculados utilizando a fórmula (3), sendo “*I*” o valor de influência para cada fator de complexidade técnica. Estes valores de impacto para cada fator devem estar entre 0, como não presente ou não influentes, e 5, para alta influência.

³ Em inglês, *Data Functions*.

Identificação	Característica
C1	Comunicação de dados
C2	Processamento distribuído
C3	Performance
C4	Utilização do equipamento
C5	Volume de transações
C6	Entrada de dados "on-line"
C7	Eficiência do usuário final
C8	Atualização "on-line"
C9	Processamento complexo
C10	Reutilização de código
C11	Facilidades de implantação
C12	Facilidade operacional
C13	Múltiplos locais
C14	Facilidades de mudanças

Quadro 1 – Fatores de complexidade técnica.

Fonte: Symons (1988, P. 3)

$$TCF = 0.65 + 0.01 \times (\sum I_{c1} + \sum I_{c2} + \sum I_{c3} + \dots) \quad (3)$$

Porém, como o FP representa apenas o tamanho do software, sem levar em consideração aspectos como gerência e riscos do projeto, habilidade da equipe, métodos, ferramentas e linguagens, este valor deve ser multiplicado por um fator de produtividade particular de cada empresa para servir como estimativa de esforço. Este valor do fator de produtividade fica a critério dos estimadores. Embora possam utilizar, em um primeiro momento, valores de referência de outras empresas, o ideal é que se revise este fator para, com o tempo, melhorar a acurácia das estimativas.

Buglione e Ebert (2011) apresentam uma lista com doze ferramentas para realização correta, do ponto de vista gerencial, da estimativa de software. Desta lista, oito ferramentas utilizam como base Pontos de Função, e as outras, contagem de Linhas de Código.

2.2.6 Rede Bayesiana

Redes Bayesianas oferecem a vantagem de serem capazes de raciocinar na presença de incertezas, suposições prévias, e dados incompletos. Podem misturar julgamento do especialista, distribuições estatísticas, e observações em um modelo único (Hearty et al. 2009).

Hearty et al. (2009) apresentam a aplicação da técnica de Redes Bayesianas para predição de velocidade da execução de iterações para projetos realizados seguindo a metodologia de desenvolvimento ágil *Extreme Programming*. Embora não tenham sido realizados testes com conjuntos de dados públicos, pelo conjunto de dados não ser acessível a outros autores e não terem realizado comparações com outros métodos de estimativa, não há como saber se haveriam métodos com melhor desempenho para o caso abordado. E segundo o que autores como Mendes e Mosley (2008) afirmam, poderiam ser encontrados melhores resultados ao utilizarem outros métodos mais apropriados para problemas desta natureza, como métodos baseados em Regressão.

2.2.7 Redes Neurais Artificiais

Redes neurais artificiais são úteis quando as distribuições não são simples funções lineares. Uma rede neural artificial contém uma camada de entrada com, por exemplo, os detalhes do projeto, zero ou mais camadas ocultas, e a camada de saída com, por exemplo, a estimativa de esforço. As conexões são arestas dirigidas ponderadas. Se o sinal que chega a um nó soma mais que um valor limite, este nó dispara, propagando um peso através da rede. O aprendizado em redes neurais artificiais ocorre comparando o valor de saída com o valor esperado, e então aplicando algum método de correção para melhorar o peso da aresta (Kocaguneli et al. 2012b) (Kocaguneli et al. 2012a).

Radial Basis Function Network (RBFN) é um tipo especial de rede artificial neural, baseado na ideia de campos receptivos biológicos. RBFN é uma rede de três camadas *feedforward*, contendo uma camada de entrada, uma camada oculta que

contém múltiplos neurônios com funções de transferência gaussiana simétrica radial, e uma camada linear de saída. A RBFN tem sido recentemente aplicada no domínio de estimativa de esforço de software (Dejaeger et al. 2012).

Já *Generalized Regression Neural Network* é um tipo especial de RBFN e, em redes deste tipo, a camada oculta contém apenas um neurônio para cada exemplo de entrada apresentado ao algoritmo durante o treinamento (Dejaeger et al. 2012).

2.2.8 Regressão

Métodos baseados em Regressão expressam as relações entre variáveis dependentes e independentes como uma função explícita conhecida com parâmetros desconhecidos, chamados de coeficientes de regressão. Esses parâmetros são estimados de modo que minimize o critério predefinido (Mittas e Angelis 2013). *Ordinary Least-Squares Regression* (OLS) minimiza a soma dos quadrados dos resíduos. *Least Median of Squares Regression* (LMS) minimiza a mediana dos quadrados dos resíduos. *Least Trimmed Squares Regression* (LTS) minimiza a soma dos quadrados dos resíduos sobre um subconjunto de pontos. *Robust M-estimator Regression* (RobhMM) minimiza um estimador M.

Dejaeger et al. (2012) afirmam que técnicas de aprendizagem de máquina funcionam melhor com conjuntos de dados pequenos e relevantes, do que com conjuntos de dados grandes e com ruídos, além de indicar atributos como linguagem de programação, tamanho da equipe e setor da companhia por se provarem bons atributos para estimativa de esforço. Dejaeger et al. (2012) sugerem também que técnicas de mineração de dados podem contribuir para o processo de estimativa de esforço como um complemento ao julgamento do especialista, determinando um multimétodo, assim como explorado por Kocaguneli et al. (2012a).

Menzies et al. (2013) e Kitchenham et al. (2007) discutem o dilema entre a aplicação de conjuntos de dados e técnicas já estabelecidos e a mineração de dados e implementação das técnicas específicas para seu contexto de aplicação. Esta pesquisa, através de uma revisão da literatura focando em estimativa de software e predição de defeitos, apresentou a análise de 4 e 28 artigos, para

respectivamente, a estimativa de software e a predição de defeitos. Para os experimentos, foram agrupadas as instâncias pelas dimensões de maior variabilidade para analisá-las em um contexto de grupo, como meio-termo a contexto local e global. Os resultados encontrados foram bastante positivos para o uso de instâncias de contexto de grupo, comparado aos outros contextos. O que corrobora para a conclusão de que métodos baseados em modelos, que aprendem não somente do conjunto de dados do contexto no qual serão aplicados, mas também de conjuntos de dados de contextos similares, obtém melhores resultados.

Para Dejaeger et al. (2012), que em seu artigo comparam a performance das abordagens de estimativa de software baseadas em mineração de dados “*Ordinary Least Squares Regression (OLS)*”, “*OLS Regression With Log Transformation (LMS)*”, “*OLS Regression With Box Cox (BC) Transformation*”, “*Robust Regression*”, “*Ridge Regression*”, “*Least Median Squares Regression*”, “*Multivariate Adaptive Regression Splines (MARS)*”, “*Classification And Regression Tree (CART)*”, “*Model Tree*”, “*Multilayered Perceptron Neural Network*”, “*Radial Basis Function Networks*”, “*Case-Based Reasoning*”, e “*Least Squares Support Vector Machines*”, a técnica que obteve o melhor desempenho foi a OLS, por obter os melhores resultados em sete dos nove casos, concordando com os resultados encontrados por Mittas e Angelis (2013) em seus experimentos. Como métricas de comparação das técnicas de estimativa, utilizaram MdMRE, Pred(25) e correlação. Porém, mesmo a melhor das técnicas comparadas, apresentou diferença entre esforço estimado mediano e esforço real entre 27,5% e 60,8%, o que ainda se mostra insatisfatório (Dejaeger et al. 2012).

Mendes e Mosley (2008), em seu artigo compara a performance de Redes Bayesianas Orientadas a Dados, Redes Bayesianas Híbridas, Analogia, Regressão, Esforço Médio e Esforço Mediano. Utilizando o conjunto de dados Tukutuku e comparando os resultados encontrados utilizando as métricas, Magnitude de Erro Relativo Média (MMRE), Magnitude de Erro Relativo Mediano (MdMRE), porcentagem de predições falhando em 25% do valor real (PRED(25)), EMRE Médio (MEMRE) e EMRE Mediano (MdEMRE). E assim como Dejaeger et al. (2012) e Mittas e Angelis (2013), obteve como a abordagem mais eficiente a Regressão, seguido por Redes Bayesianas Híbridas e Analogia.

Já Mittas e Angelis (2013), em seu artigo, no qual apresentam uma metodologia estatística para comparação de modelos de predição para estimativa de

software, utilizam 6 conjuntos de dados para comparar 11 modelos de predição, com o método de validação *k-Fold Cross-Validation*, e apresentam como método com o melhor desempenho o *Ordinary Least-Squares Regression (OLS)* e o *Classification And Regression Tree (CART)* como o com menor desempenho para realização de estimativas de esforço, discordando de Kocaguneli et al. (2013), que apresentou CART como com melhor desempenho. Porém a falta do uso de *Median of the Magnitude of Relative Error (MdMRE)* como medida de taxa de erro, e a não utilização de *Leave-One-Out Cross-Validation* como estratégia de validação, podem trazer resultados diferentes, como encontrado em (Kocaguneli et al. 2012b).

2.2.9 Outros

Há modelos de estimativa baseados em teoria, como por exemplo, o modelo de estimativa SLIM e modelos de estimativa baseados em “ciência do software” (Jørgensen e Shepperd 2007).

Sobre simulação, Suelmann (2013), em sua simulação de Monte Carlo, contradiz os resultados encontrados por Symons (2012). O primeiro autor expõe o erro de derivação ocorrido no artigo original através de cálculos matemáticos. E discorda que o modelo possa refletir a realidade, já que a duração do projeto afeta o esforço e vice-versa.

Há estudos também de modelos de estimativa baseados em *Work Break-Down (WBS)* ou baseados em decomposição de atividades (Jørgensen e Shepperd 2007).

Outros métodos não muito discutidos são *General Linear Model (GLM)*, *Genetic Programming (GP)*, e *Stepwise ANOVA*, que foram utilizados para comparação na revisão sistemática da literatura de Kitchenham et al. (2007), com bons resultados.

Já Mendes e Mosley (2008), em suas comparações, utilizaram Esforço Médio e Esforço Mediano que, por serem muito simples de serem realizados e pouco adaptativos, geralmente não são vistos como métodos de estimativa, mas sim como fórmulas de sumarização de resultados de outros métodos de estimativa. E enquanto Esforço Mediano teve um bom resultado, Esforço Médio ficou entre os

menos acurados.

2.3 PLANNING POKER

Esta técnica de estimativa é bastante utilizado em métodos ágeis para o desenvolvimento de software, especialmente com Scrum e *Extreme Programming* (Mahnič e Hovelja 2012). Foi introduzido por Grenning (2002) e popularizado por Cohn (2005).

Segundo Raith et al. (2013), o fluxo de execução do *Planning Poker*, exibido na Figura 1, envolve todos os membros do time de desenvolvimento como estimadores, um moderador (*Scrum Master*, por exemplo) e, opcionalmente, um representante do cliente.

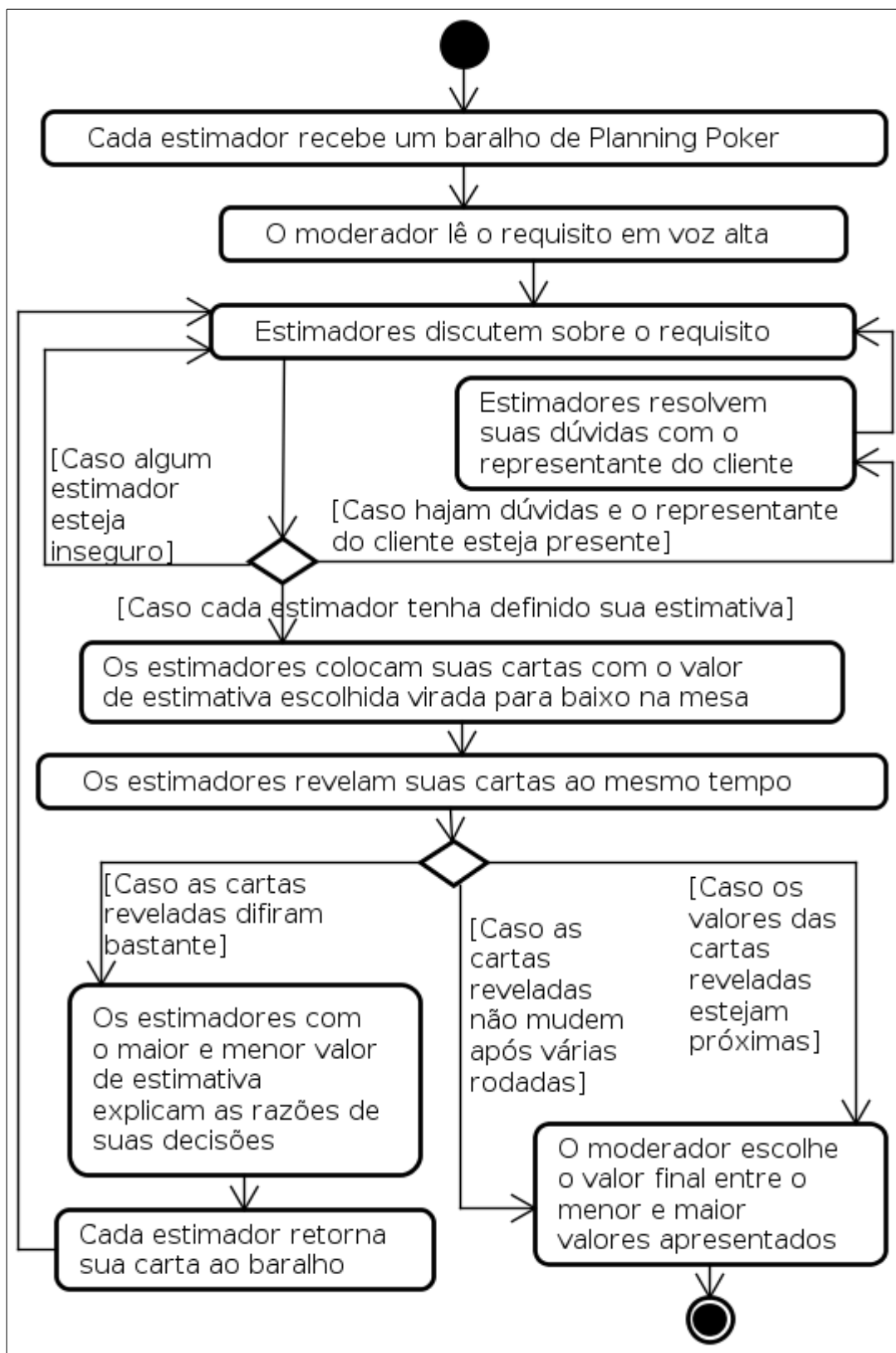


Figura 1 – Diagrama do fluxo de execução do *Planning Poker*.

Conforme visualizado na Figura 1, o *Planning Poker* ocorre como um jogo no

qual o moderador lê um requisito, os três ou mais participantes (estimadores) discutem sobre o requisito, podendo tirar dúvidas quanto ao requisito com um representante do cliente, se este estiver presente. Cada participante coloca uma carta na mesa com valor da estimativa virada para baixo, sendo que todas são reveladas simultaneamente, quando todos os participantes tiverem colocado as suas cartas na mesa. Se os valores das estimativas estiverem muito diferentes, os participantes com maior e menor valor de estimativa devem se justificar, as cartas retornam ao baralho e as estimativas são colocadas na mesa novamente até que os participantes cheguem a um consenso, ou várias rodadas ocorram sem mudança nos valores apresentados, e o processo termina com o moderador selecionando a estimativa que acredita ser a mais próxima entre o menor e maior valores apresentados.

Esta técnica de estimativa de esforço de software se destaca por utilizar tanto o conhecimento técnico de desenvolvimento de software, como a capacidade de prever o software ainda a ser desenvolvido (Børte et al. 2012).

Outro fator positivo do uso da técnica de estimativa Planning Poker é que diversos especialistas compartilham diferentes ideias e perspectivas, sendo realmente útil para pensar nas diferentes condições que podem ocorrer durante a implementação dos requisitos (Desharnais et al. 2011). E, segundo Desharnais et al. (2011), diferente de outras técnicas de estimativa, pelo fato do processo do Planning Poker ser basicamente um jogo, não se torna cansativo.

Segundo Williams et al. (2011), Planning Poker provê um meio estruturado para:

- Obter a compreensão do requisito por todos os participantes.
- Expor suposições ocultas de aspectos técnicos de implementação e verificação.
- Discutir as implicações da implementação do requisito no sistema.
- Descobrir e resolver ambiguidades através de perspectivas divergentes sobre o requisito.
- Expor alternativas para a implementação do requisito.

Planning Poker é visto como uma técnica que provê melhores estimativas, comparadas a estimativas individuais, ao permitir que todos os estimadores participem, independente de sua capacidade de persuasão (Mahnič e Hovelja 2012),

além de incentivar a interação social durante a construção de estimativas (Børte et al. 2012).

Segundo Børte et al. (2012), *Planning Poker* é interessante por:

- Ser organizado de modo a permitir que todos participem através de suas cartas, independente da posição ou experiência.
- Requisitar justificativas para os participantes que estimarem com o maior e o menor valor, como parte do processo de estimativa.
- Utilizar as cartas de jogo como artefatos de realização de estimativa.
- Permitir, de modo organizado, que especialistas de diferentes áreas estimem juntos, auxiliando a compreensão da solução de problemas altamente complexos.

Segundo os estudos de Raith et al. (2013), o processo necessário para realização do *Planning Poker* oferece estimativas mais acuradas e menos otimistas comparado também com a média das estimativas, além das estimativas realizadas em grupo resultarem em estimativas mais acuradas que estimativas individuais.

O uso de *Planning Poker* reduz o otimismo nas estimativas quando comparadas a combinação das estimativas individuais (Moløkken-Østvold et al. 2008) (Raith et al. 2013) (Williams et al. 2011). Porém, quando realizados por participantes inexperientes, para requisitos semelhantes e valores extremos, tiveram seus desvios intensificados (Raith et al. 2013).

A interação cara a cara pode ser um fator que faz com que os participantes tomem mais cuidado com suas decisões (Moløkken-Østvold et al. 2008).

Em teoria, *Planning Poker* possui diversas características que deveriam proporcionar melhoras na acurácia das estimativas de esforço por combinar conhecimento de diversos profissionais, poderem ser realizadas com métodos de desenvolvimento iterativo, e por revelarem as estimativas ao mesmo tempo, reduzindo o viés social da comparação (Moløkken-Østvold et al. 2008).

Segundo Peixoto et al. (2010), 12% dos 551 profissionais que responderam sua pesquisa responderam que utilizam *Planning Poker* como principal técnica de estimativa. Porém apenas três quartos destes aplicam esta técnica de forma adequada.

Moløkken-Østvold et al. (2008), citam Moløkken-Østvold e Jørgensen (2004), onde estes descrevem que as estimativas em grupo realizadas após uma discussão

não estruturada eram menos otimistas e mais realistas que estimativas individuais, realizadas antes da discussão, combinadas. A principal fonte dessa diminuição de otimismo aparentou ser a identificação de atividades adicionais e a percepção de que atividades eram mais complexas do que inicialmente pensado.

Børte et al. (2012) e Moløkken-Østvold et al. (2008), citam Haugen (2006), que comparou o uso de *Planning Poker* com a realização de estimativas em grupo não estruturado para investigar o impacto no desempenho da estimativa. Com um total de 101 requisitos estimados em ambiente de desenvolvimento XP (*Extreme Programming*), foi encontrado aumento da acurácia de estimativas para atividades familiares ao comparar com o grupo não estruturado. Porém, com diminuição da acurácia de estimativas ao estimar atividades não familiares.

Segundo Moløkken-Østvold et al. (2008), um problema frequentemente reportado na área de pesquisa é que experimentos tendem a investigar atividades e/ou resultados hipotéticos, e não atividades executadas realmente com seus resultados registrados.

Embora a técnica de estimativa *Planning Poker* não seja tão confiável, por se basear em estimativas humanas e por se não utilizar de análises como medidas históricas ou analogia, uma abordagem para prevenir estes problemas é utilizar um repositório histórico (Desharnais et al. 2011).

Moe et al. (2012) aponta a falta de conhecimento relacionado ao requisito, falta de tempo para preparação e análise, e falta de comunicação entre os integrantes do grupo de estimadores como causadoras de estimativas que subestimam a complexidade da implementação do requisito em ambientes de decisão compartilhada.

Não há posição clara da área de pesquisa sobre os valores das cartas usadas no *Planning Poker*, pois mesmo sendo definido por Grenning (2002) com os valores “1”, “2”, “3”, “5”, “7”, “10” e “infinitos” dias, Cohn (2005), que popularizou a técnica, definiu os valores como “0,5”, “1”, “2”, “3”, “5”, “8”, “13”, “20”, “40” e “100” dias. Já Dagnino (2013), definiu os valores como “1”, “2”, “3”, “5”, “8”, “13”, “20”, “30”, “40”, “50”, “70”, “90” e “100” dias no melhor caso, convertendo as respostas em valores menos otimistas. Grenning (2002) permite ainda o uso de duas cartas para definir valores intermediários, além de definir que o requisito deverá ser dividido em requisitos mais simples caso uma carta “infinito” seja jogada.

2.4 EDUCAÇÃO E TREINAMENTO EM ANÁLISE DE REQUISITOS

Ambientes computacionais para apoio ao ensino relacionado a requisitos de software tem sido desenvolvidos com diferentes vertentes. Santos et al. (2013) desenvolveram um jogo para ensino de conceitos de requisitos em sistemas ubíquos. Zuppiroli, Ciancarini e Gabrielli (2012) desenvolveram um RPG (*role-playing game* ou jogo de interpretação de papéis) que permite que requisitos possam ser discutidos e avaliados em um laboratório de engenharia de software. Thiry, Zoucas e Gonçalves (2010) criaram um jogo educativo que faz uso de aspectos lúdicos e de desafios para promover a aprendizagem da Engenharia de Requisitos. Vargas et al. (2010) desenvolveram um jogo para a aprendizagem de elicitação de requisitos de software, através do uso de situações e cenários simulados.

Em relação ao acompanhamento do desempenho de alunos durante o processo de aprendizado em projeto de sistemas de software, a ferramenta SEREBRO fornece o desempenho de grupos no desenvolvimento de projetos de software, analisando a colaboração, a contribuição e o progresso de cada um (Hale, Jørgensen e Gamble 2011).

Práticas e ferramentas para apoio ao ensino de Engenharia de Software, sem o uso de ambientes computacionais, também tem sido criadas, como o jogo AprendES (Feitosa e Campos 2010). Esse é um jogo baseado em cartas, que permite estimar o esforço para se construir um projeto, definindo-se o número de módulos do projeto utilizando-se cartas do tipo 'tamanho'. Já a ferramenta Modelando (Silva et al. 2012) é um jogo educacional para o ensino de Engenharia de Requisitos que proporciona aos estudantes da disciplina um exercício prático, lúdico e iterativo com o objetivo de praticar os conceitos assimilados, utilizando-se os diagramas da UML (Linguagem de Modelagem Unificada).

Porém, nenhum dos ambientes de apoio ao aprendizado citados anteriormente envolve a análise dos requisitos para estimativa de esforço utilizando *Planning Poker*. Também não utilizam requisitos e esforços reais para apoio ao aprendiz de estimativa de software.

3 REVISÃO DE ATIVIDADES EXECUTADAS

Como a técnica de estimativa de software baseada em julgamento de especialista *Planning Poker* se baseia no uso da experiência do estimador para elaboração da estimativa, este método não segue um método formal definido.

O julgamento do especialista, por ser realizado por seres humanos, sofre influências não apenas das características das atividades ou projetos a serem estimados, como prazo, complexidade, tamanho, método de desenvolvimento, entre outros. Essas outras influências, como as exploradas por Jørgensen e Grimstad (2012) e Jørgensen (2013b), podem ser estudadas separadamente para se realizar um apanhado destas, definirmos práticas que auxiliem o estimador a conseguir resultados mais acurados.

Hoje, na literatura, o contexto de estimativa de software baseado em julgamento de especialista é muito amplo e informal. Por isso a necessidade de focar em um método de execução de estimativas com revisão de atividades executadas.

Para verificar uma possível melhora no uso de revisões de atividades executadas, foram alterados três dos passos definidos no diagrama da Figura 1, como destacado no diagrama da Figura 2.

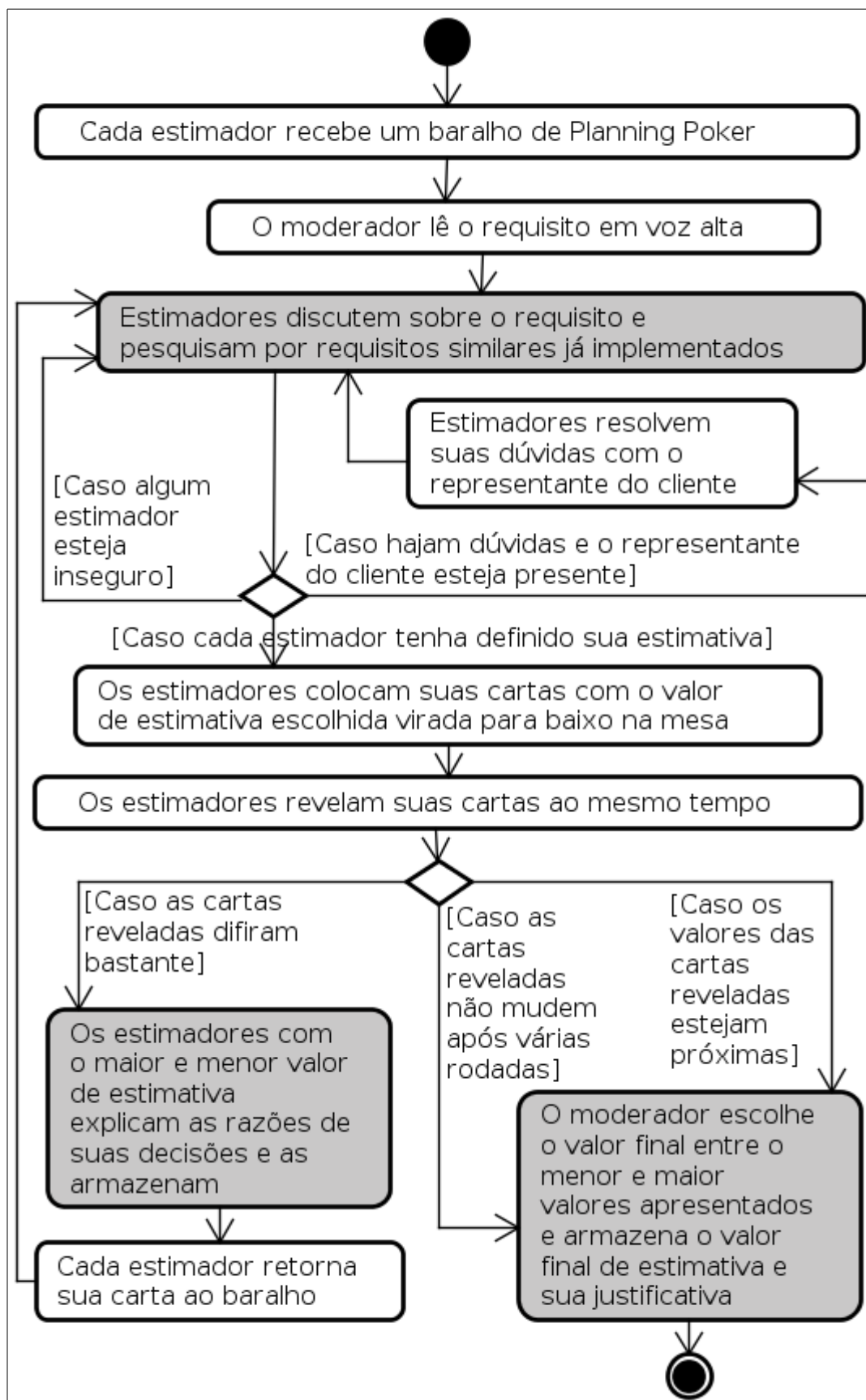


Figura 2 – Diagrama do fluxo de execução do *Planning Poker* com revisão de atividades executadas.

4 MATERIAIS E MÉTODOS

Este capítulo descreve como foram realizados os quatro experimentos.

O método de aplicação dos experimentos foi definido para este trabalho e está descrito nas Figuras 3, 4, e 5, para as duas iterações, com 79 alunos da Universidade Tecnológica Federal do Paraná (UTFPR) dos cursos Bacharelado em Sistemas de Informação e Engenharia de Computação.

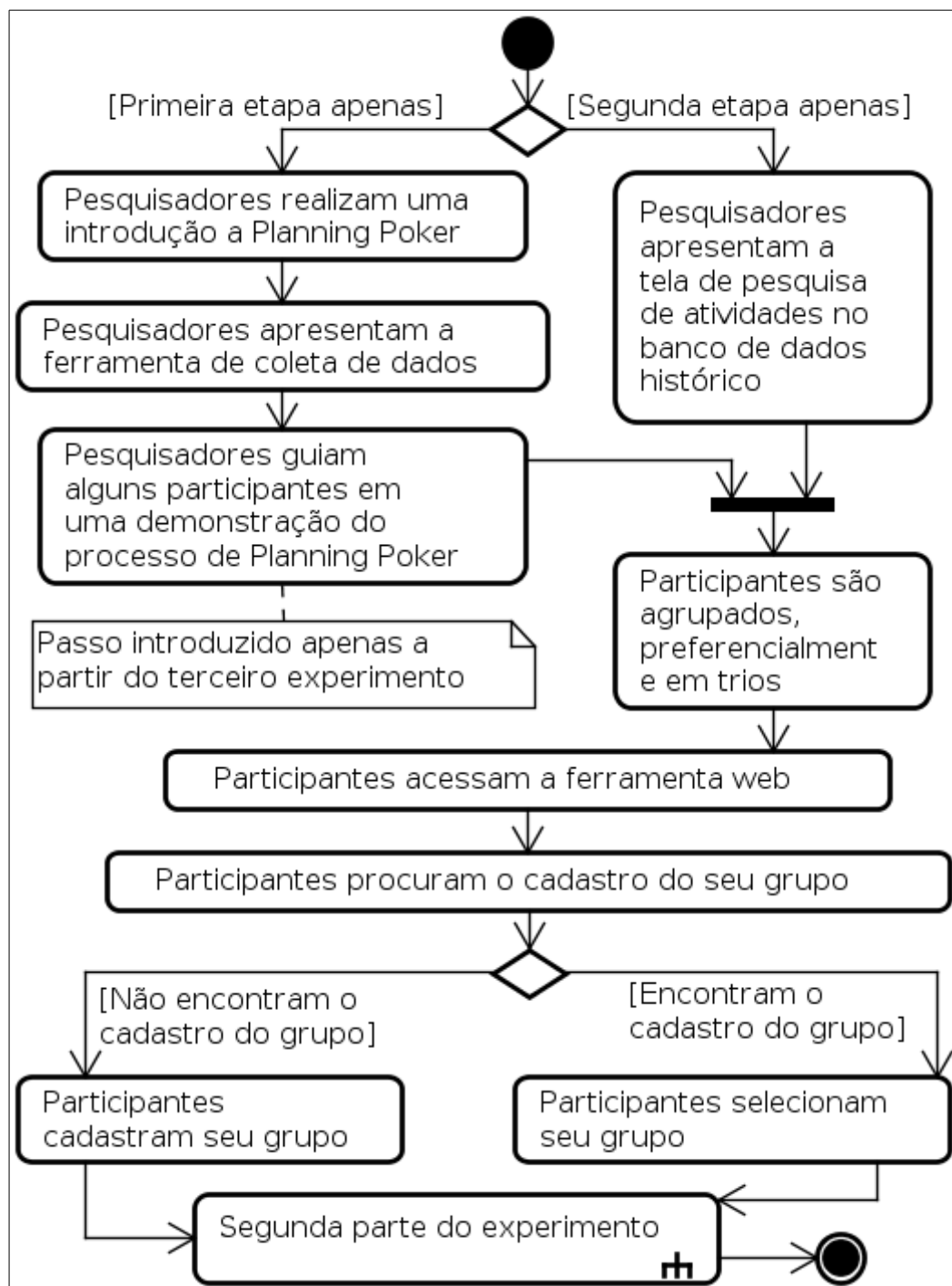


Figura 3 – Diagrama do fluxo de execução do experimento – primeira parte.

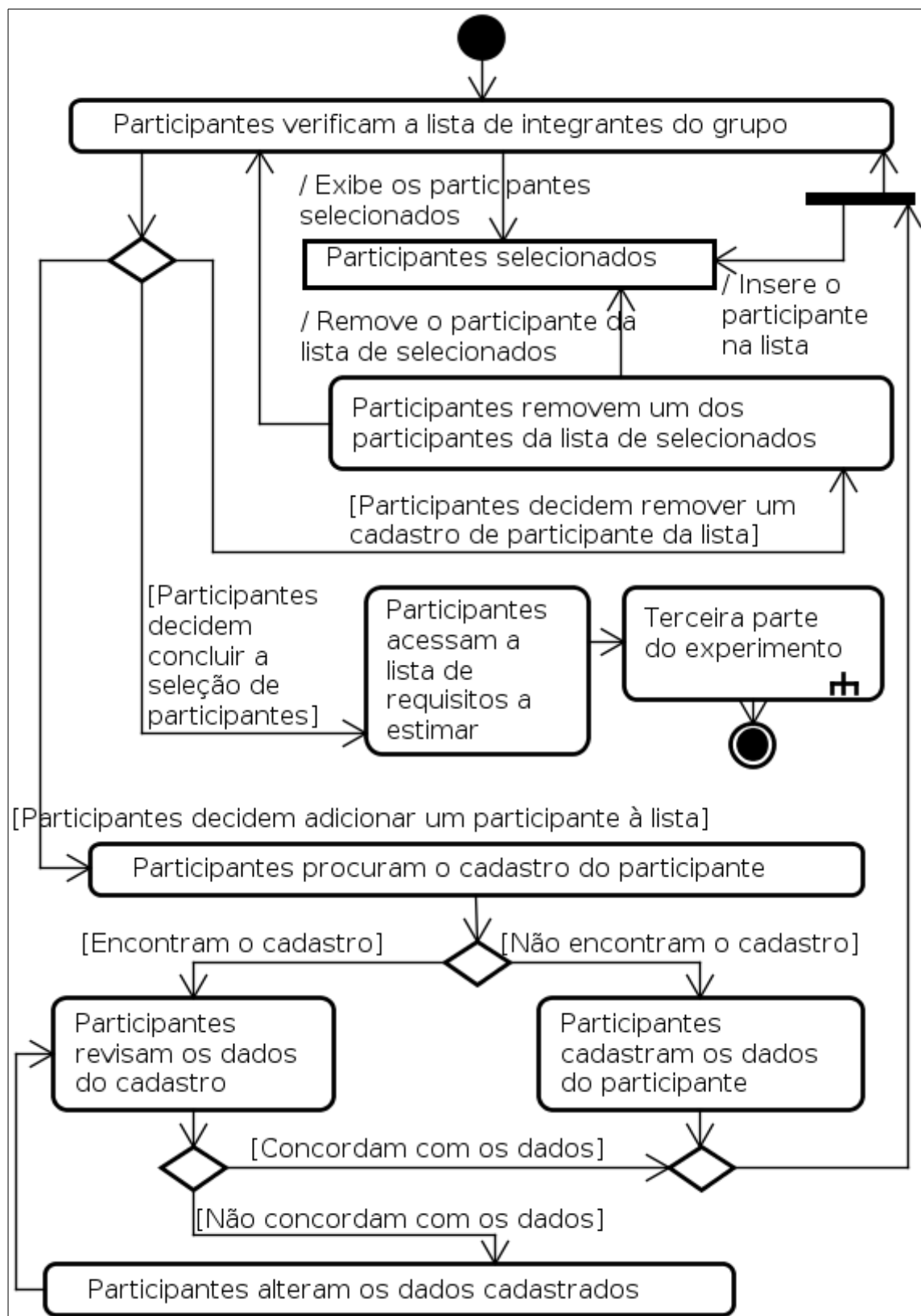


Figura 4 – Diagrama do fluxo de execução do experimento – segunda parte.

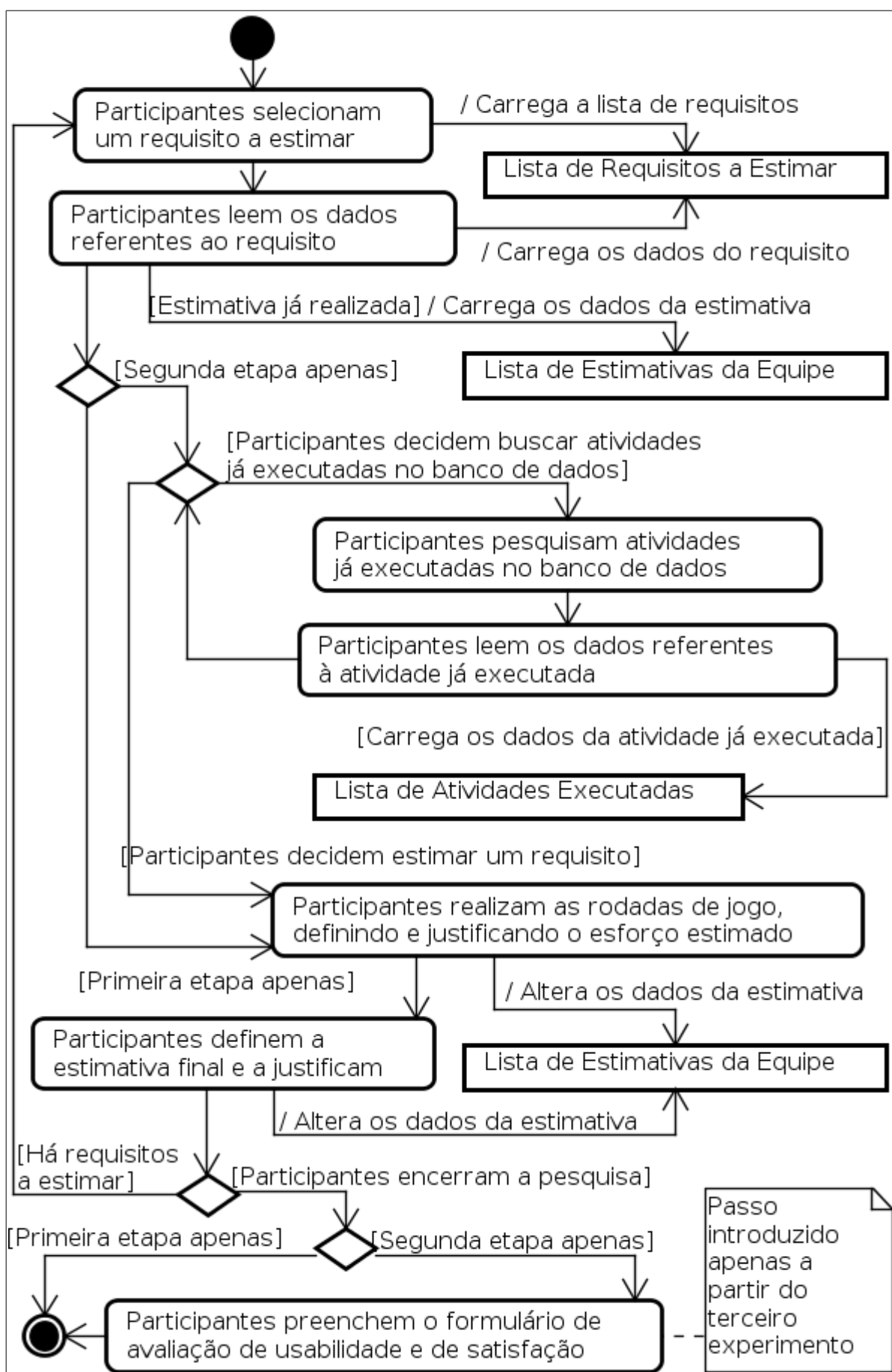


Figura 5 – Diagrama do fluxo de execução do experimento – terceira parte.

Na primeira etapa do experimento, cada participante integrante do grupo de estimativa deve se cadastrar, inserindo os seus dados pessoais e as suas competências profissionais e acadêmicas, conforme indicado pela atividade “Participantes cadastram os dados do participante” do diagrama da Figura 4. Os participantes cadastrados podem fazer a atualização dos dados cadastrados, se desejarem, como indicado na atividade “Participantes revisam os dados do cadastro” do diagrama da Figura 4.

Após o cadastro de todos os participantes, é possível selecionar, a partir de uma lista de requisitos, cada um dos requisitos e as suas especificações, a fim de ser feita a estimativa de esforço em horas-homem para o mesmo. Esta descrição é relativa à atividade “Participantes leem os dados referentes ao requisito” do diagrama da Figura 5.

Ao preencherem os campos de valor de estimativa e de justificativa para cada estimador, como na Figura 5, e terminarem a rodada, os dados respectivos a esta rodada são armazenados, e a tela é atualizada para permitir uma nova rodada, exibindo os valores anteriores estimados como “Valor Anterior” para que o moderador possa saber se os estimadores estão mudando suas estimativas ou não estão chegando a um consenso. Esta ação executa a atividade “Participantes realizam as rodadas de jogo, definindo e justificando o esforço estimado” do diagrama da Figura 5.

Caso os participantes cheguem a um consenso ou o moderador decida encerrar as rodadas por falta de consenso, os participantes devem definir o valor da estimativa final, permitindo a inserção da estimativa final. Os participantes devem, então, preencher os campos de estimativa final e de razões e incertezas. Esta ação executa a atividade “Participantes definem a estimativa final e a justificam” do diagrama da Figura 5. As razões e incertezas podem estar relacionadas a fatores como complexidade, dificuldade da linguagem de programação, ou profundidade insuficiente dos requisitos.

Apenas na segunda etapa, a fim de verificar o impacto da utilização da busca em repositório histórico durante a execução de estimativas utilizando *Planning Poker*, os alunos passam a ter acesso a buscas por palavras-chave relativas, entre outros filtros, aos requisitos a serem estimados. Esta ação executa a atividade “Participantes buscam atividades já executadas no banco de dados” do diagrama apresentado na Figura 5.

4.1 TERMOS UTILIZADOS

Para melhor descrever os passos realizados, foram definidos alguns termos. Este capítulo do trabalho descreve alguns destes termos.

Cada execução do processo de aplicação de experimento, como descrito no diagrama das Figuras 3, 4, e 5, é chamada de experimento por poder ser analisada isoladamente.

Cada experimento contém duas etapas similares em todos os aspectos, inclusive com as equipes formadas com os mesmos integrantes, se possível. Porém com a diferença de proporcionar acesso ao banco de dados histórico na segunda etapa.

Por conta da necessidade de amadurecimento da ferramenta de coleta e do método de aplicação dos experimentos, foram divididos os experimentos em primeira iteração, contendo os primeiros dois experimentos, e segunda iteração, contendo os experimentos seguintes.

Cada grupo de participantes, ao realizar a técnica *Planning Poker*, foi chamado de time, por terem de resolver os problemas juntos.

4.2 TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Para obter permissão dos participantes para utilizar os dados gerados por eles nesta pesquisa, foi elaborado o documento presente no Apêndice B.

4.3 QUESTIONÁRIO DE AVALIAÇÃO DA FERRAMENTA

Para compreender o ponto de vista do participante nos quesitos usabilidade e satisfação, foi elaborado um questionário, presente no Apêndice C, e aplicado no fim dos experimentos da segunda iteração.

4.4 CARTAS POSSÍVEIS

Embora seja mais comum utilizar cartas cujos valores já estejam impressos nestas, para evitar a perda de acurácia com grandes valores, começando com metade de um dia (Cohn 2005) ou começando com um dia (Grenning 2002) (Dagnino 2013), por serem utilizados requisitos simples, de em média “1,35” horas e mediana de “0,85” horas, não foi vista necessidade de limitar os valores, permitindo assim que os participantes usem valores baixos, mais interessantes pela baixa complexidade dos requisitos.

4.5 ANÁLISE ESTATÍSTICA

Para investigar possíveis influências na acurácia das estimativas realizadas foi escolhida a fórmula de coeficiente de correlação de Pearson.

Como índice de desempenho acadêmico, cuja relação com o melhor uso da ferramenta foi questionada, foi utilizada a média das notas dos alunos em disciplinas de computação.

Os valores a serem verificados para buscar correlações foram definidos como:

- Média das notas dos participantes em matérias de computação e média do aumento relativo (%) da média da diferença absoluta.
- Média das notas dos participantes em matérias de computação e o número de estimativas que melhoraram.
- Média das notas dos participantes em matérias de computação e as médias da média da diferença absoluta da etapa 1.
- Média das notas dos participantes em matérias de computação e as médias da média da diferença absoluta da etapa 2.
- Média das notas dos participantes em matérias de computação e o número de estimativas que pioraram.
- Número de participantes e a média dos valores da média da diferença absoluta da etapa 1.

- Número de participantes e a média dos valores da média da diferença absoluta da etapa 2.
- Número de participantes e média do aumento relativo (%) da média da diferença absoluta.
- Número de participantes e o número de estimativas que melhoraram.
- Número de participantes e o número de estimativas que pioraram.

4.6 MODELAGEM DA FERRAMENTA DE COLETA DE DADOS DO EXPERIMENTO

A ferramenta de software foi idealizada neste trabalho para coletar os dados e metadados gerados pelos participantes em cada etapa do *Planning Poker*, para assim poder buscar correlações entre seus resultados e como estes foram informados.

4.6.1 Banco de Dados Legado

Os dados de projetos e requisitos, incluindo o tempo gasto para implementá-los, foram retirados do software de planejamento e monitoramento de execução de atividades de desenvolvimento de software, chamado *Projects*, utilizado em um ambiente real de fábrica de software. Das atividades disponíveis no banco de dados legado, do qual foram importados e processados os dados, apenas os requisitos funcionais foram selecionados, de modo a tornar menos abstratas as alterações necessárias, e assim facilitar a compreensão pelos participantes.

4.6.2 Diagrama de Classes

O diagrama, exibido na Figura 6, foi definido após investigar o processo de *Planning Poker* e analisar os dados do banco legado do ambiente real de

desenvolvimento, que permitiu, respectivamente, levantar os dados a coletar no experimento e os que poderiam alimentar inicialmente a ferramenta. A extração, transformação e carga dos dados é descrito com mais detalhes no Capítulo 5.1.

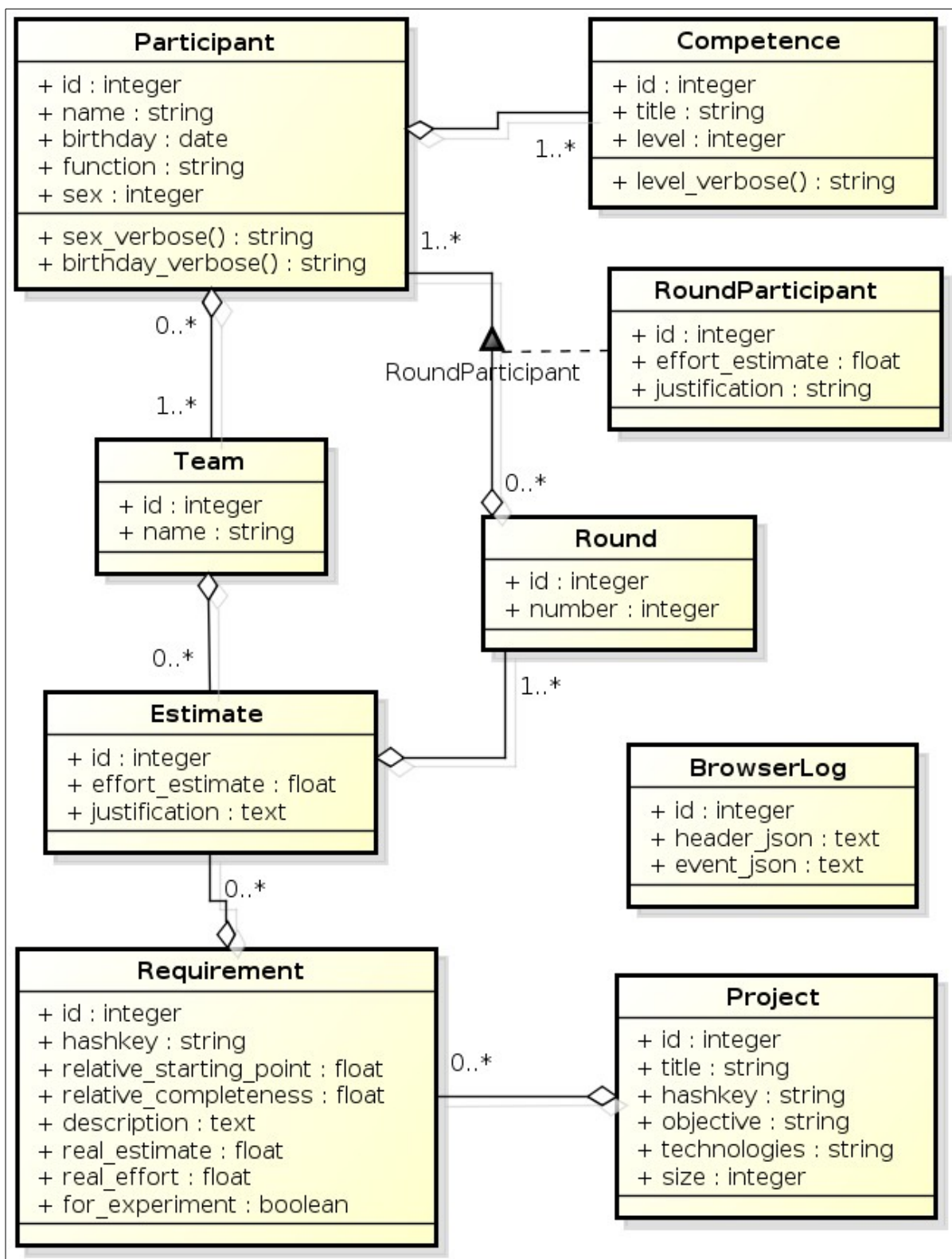


Figura 6 – Diagrama de classes da primeira versão da ferramenta de coleta de dados dos experimentos.

4.6.3 Rascunhos de Tela

Rascunhos de tela são utilizados na etapa de projeto ou análise de software, bastante úteis por serem facilmente compreendidos pelos envolvidos no projeto, embora sejam bastante limitados no que podem apresentar.

Para desenvolver uma ferramenta que auxiliasse os participantes a executarem o *Planning Poker*, com fluxo definido na Figura 1, e que permitisse a realização dos experimentos, como definido nas Figuras 3, 4, e 5, se fez necessário rascunhar as telas, imaginando como as interações dos usuários ocorreriam.

4.6.3.1 Cadastro de Dados do Participante

Além de identificar o participante, foram escolhidos para entrada de dados o nome completo, a data de nascimento, função que desempenha como profissional, sexo e competências que os destacam, como apresentado na Figura 7. Este rascunho de tela ilustra a interface utilizada para as atividades relativas à manutenção dos dados dos participantes, do diagrama da Figura 4, como “Participantes verificam a lista de integrantes do grupo”, “Participantes removem um dos participantes da lista de selecionados”, “Participantes procuram o cadastro do participante”, “Participantes cadastram os dados do participante”, “Participantes revisam os dados do cadastro”, e “Participantes alteram os dados cadastrados”.

Coletor de dados da estimativa

← → × ↶

Nome	Adicionar
Participante 1	Editar Excluir
Participante 2	Editar Excluir

Participante 1

Nome Completo:

Data de Nascimento:

Competência	Básico	Intermediário	Avançado	
Java	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
PHP	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Remover
SQL	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text" value="C++"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Adicionar

Caso trabalhe, função que desempenha:

Sexo: Masculino Feminino

Figura 7 – Rascunho de tela de tela de cadastro de dados do participante.

4.6.3.2 Análise de Requisitos e Definição de Estimativa Final

Como os dados interessantes disponíveis já haviam sido definidos durante o processo de coleta de dados legados, a importância desta tela foi definir a apresentação dos dados, e a forma de entrada da estimativa final dada pelos participantes, como apresentado na Figura 8. Este rascunho de tela ilustra a interface utilizada para as atividades relativas à exibição dos dados do requisito e para as atividades relativas ao cadastro da estimativa final, do diagrama da Figura 4, como “Participantes acessam a lista de requisitos a estimar”, e do diagrama da Figura 5, como “Participantes leem os dados referentes ao requisito”, e “Participantes definem a estimativa final e a justificam”.

Coletor de dados da estimativa

Nome

Requisito 1	Abrir
Requisito 2	Abrir
Requisito 3	Abrir

Requisito 1

Objetivo do projeto: Criação de ferramenta para gerenciamento de eventos

Tecnologias empregadas: PHP, MySQL, Javascript.

Tamanho do projeto estimado: 400 homens-hora

Momento do projeto:

Quanto já foi entregue:

Descrição: Adicionar área de "fale conosco" com os campos nome, e-mail e mensagem. E ao cadastrar esta mensagem no banco de dados, deve ser enviado um e-mail para administrador@sistema.com.br. Uma tabela no banco de dados deverá ser criada para armazenar as mensagens cadastradas.

Estimativa:

Descreva livremente, e com o máximo de detalhes a razão de ter (em) chegado a esta estimativa:

Figura 8 – Rascunho de tela de análise de requisitos e definição de estimativa final.

4.6.3.3 Cadastro de Estimativas Individuais por Rodada

Para respeitar o processo de estimativa do *Planning Poker*, apresentado na Figura 1, e coletar os dados gerados neste processo deve se permitir a entrada de estimativas individuais para as rodadas realizadas, como apresentado na Figura 9. Este rascunho de tela ilustra a interface utilizada para as atividades relativas ao cadastro das estimativas individuais por rodada do *Planning Poker*, do diagrama da Figura 5, como “Participantes realizam as rodadas de jogo, definindo e justificando o esforço estimado”.

Nome	Valor Anterior	Valor Atual	Justificativa
Participante 1	3	4	...
Participante 2	4	6	...
Participante 3	6	8	...

Terminar rodada

Figura 9 – Rascunho de tela de cadastro de estimativas individuais para cada rodada do *Planning Poker*.

4.6.3.4 Busca no Repositório de Atividades Executadas

Para permitir a busca das atividades executadas, acessível apenas na segunda etapa de cada experimento, foram definidos os campos simples opcionais de filtro “Descrição”, cujo conteúdo textual deve estar contido no campo de descrição da atividade executada a exibir, “Projeto”, com uma lista para seleção de um único projeto, descritos por nome e tamanho do projeto, e campos numéricos “Esforço estimado mínimo”, “Esforço estimado máximo”, “Esforço realizado mínimo”, e “Esforço realizado máximo”, como apresentado na Figura 10. Este rascunho de tela ilustra a interface utilizada para as atividades relativas a busca no repositório de atividades executadas, do diagrama da Figura 5, como “Participantes buscam atividades já executadas no banco de dados” e “Participantes leem os dados referentes à atividade já executada”.

Coletor de dados da estimativa

← → × ↶ 🔍

Filtros de Busca

Descrição:

Projeto:

Esforço estimado mínimo: Esforço estimado máximo:

Esforço realizado mínimo: Esforço realizado máximo:

Projeto	Momento de Início de Execução	Esforço Estimado	Esforço Realizado	Descrição	
Projeto Z	23%	3	5	Implementação do relatório de acessos na...	Abrir
Projeto Z	64%	4	5	Implementação das telas de cadastro de us...	Abrir

Figura 10 – Rascunho de tela da tela de busca no repositório de atividades executadas.

5 FERRAMENTAS DESENVOLVIDAS

Para coletar, verificar e analisar os dados, foram implementadas quatro ferramentas: de migração de dados legados, de recuperação de requisições do navegador, de coleta de dados, e de análise de dados.

As ferramentas deste capítulo foram construídas utilizando-se:

- RubyOnRails 4.1.4, como *framework* de desenvolvimento *back-end*.
- jQuery 1.11.1, como *framework* de desenvolvimento *front-end*.
- MariaDB 5.5.39, como sistema gerenciador de banco de dados.
- Xubuntu 14.04 x64, como sistema operacional.
- Firefox 32 e Chrome 37, como navegadores.

5.1 FERRAMENTA DE MIGRAÇÃO DE DADOS LEGADOS PARA A FERRAMENTA DE COLETA DE DADOS

Como no software de planejamento e monitoramento de execução de atividades de desenvolvimento de software os dados eram armazenados em diversos arquivos, um para cada instância, em formato YAML cuja estrutura variava entre entidades e versões do software, foi necessária a criação de uma ferramenta de migração. Esta ferramenta foi construída especificamente para este fim. As entidades utilizadas do banco de dados de origem foram “Projeto” e “Carta”.

5.1.1 Entidade Projeto

A entidade “Projeto” foi armazenada em arquivos nomeados “project.yml”, no diretório reservado aos dados exclusivos do projeto, com atributos variáveis como apresentados nos Quadros 2, 3, e 4.

Para esta entidade foram lidos os campos “id”, para referência durante a importação, “nome”, para servir de referência aos projetos no qual o autor principal

acompanhou como parte da equipe de desenvolvimento. O campo “descrição” poderia ter sido usado para exibição ao usuário, porém não foram encontrados projetos válidos com este atributo preenchido.


```

---
ativo: false
gerente_de_projetos_id: ""
user_ids:
- 857fcd88953583704d2c9f75cc1d2845d4e5716b
- 670c924d035aa89186b290e6e3127a5055738fcc
- 8f9edb773957339c2288df4b6bc9cb898071e4eb
- d7c6f99d5e30f3110231f95994b627f2774381e5
analista_de_ppqa_id: ""
fornecedores_de_requisitos: ""
data_prevista_de_conclusao: 2013-01-31
fase_atual: ""
svn_projeto: ""
cargos: "Gerente de Projetos\r\n\
Gerente Administrativo\r\n\
Programador Efetivo\r\n\
Programador Estagi\r\n\
Analista de PPQA\r\n\
Auditor de PPQA\r\n\
Analista de CM\r\n\
Auditor de CM\r\n\
Designer\r\n\
Ilustrador\r\n\
Product Owner"
analista_de_configuracao_id: ""
svn_produto: ""
qtd_requisitos: 1
auditor_de_configuracao_id: ""
id: 5f1a85b0fccb8619340acf93b991ecd4caa23ad7
documentos_rastreabilidade: "Plano de Projeto\r\n\
Cronograma\r\n\
Planilha de Custos\r\n\
Planilha Tamanho de Projeto\r\n\
Plano de Ger\r\n\
Lista de Riscos\r\n\
Manual de Utiliza\r\n\
updated_at: 2013-04-02 20:36:23.880741 -03:00
created_at: 2012-12-19 11:47:19.411633 -02:00
nome: Projeto Teste
descricao: ""
auditor_de_ppqa_id: ""
metodos:
- "acrescentadas regras para redirecionamento de acesso e
libera\r\n\
git_pasta: ""

```

Quadro 2 – Exemplo 1 de arquivo YAML do banco de dados legado para entidade Projeto

```

---
gerente_de_projetos_id: c5832816e07274dc8b46799e19bf712a7e9c1ee1
fase_atual: ""
data_prevista_de_conclusao:
metodos:
- fim
- 12e6343a125ce87163a1785f3649b0502d8f137f
- 520401eeb8cf3e1141a4b50a9372d701590dbd53
- ok
- 203ea343f1fa973fec03c1100ace3384ee2cf56
- ...
- 24bad357f21937f11e44b91f7eb2519ddf5ed425
- !binary |
  dmVyawZpY2HDp80jbw==

- "#"
- 9d9b103fdf196622ddcef7074c1814ee73c8d937
- b3ca4e4fb61247f5b39fee3dfda36cc70aadfd3d
- aa3f1f66566174352a6fedbc1d30ef568ac94e7c
- 39aff8c01a7ca0ba167c0af8a753d5b25cb6d42b
- "n\xC3\xA3o tem"
user_ids:
- 857fcd88953583704d2c9f75cc1d2845d4e5716b
- 2a3fbb694588cc735c2b4f4832907a0c48e2ced3
- 1547efaabf881e837579449b08ff7f477b72ef5b
- 670c924d035aa89186b290e6e3127a5055738fcc
- c5832816e07274dc8b46799e19bf712a7e9c1ee1
- 06ea31ca93cae8822ed7c832b6b5ca54f922c324
- 82f11a20f35e8d19b43893fceb0490ecea884d
- 05248f01b4e961af6a45b560c55e7401633c79e9
- 8f9edb773957339c2288df4b6bc9cb898071e4eb
- 5b5cdab5eec069d8e938c463119720210d602a71
- c59e249dc67eb3507a2397b9dd263b2ce5788676
- 23e207415901b5a5f69f92e418c5c0d7bfe573d8
- 4ab8eb9428933d1f7f8ea61d35ec753b5db12beb
svn_produto: ""
fornecedores_de_requisitos: ""
cargos: "Gerente de Projetos\r\n\
  Gerente Administrativo\r\n\
  Programador Efetivo\r\n\
  Programador Estagi\xC3\xA1rio\r\n\
  Analista de PPQA\r\n\
  Auditor de PPQA\r\n\
  Analista de CM\r\n\
  Auditor de CM\r\n\
  Designer\r\n\
  Ilustrador\r\n\
  Product Owner"
descricao: ""
auditor_de_configuracao_id: ""
updated_at: 2013-05-22 14:49:14.258312 -03:00
nome: Ferramenta Y - WEB

```

Quadro 3 – Exemplo 2 de arquivo YAML do banco de dados legado para entidade Projeto

```
ativo: true
analista_de_ppqa_id: ""
documentos_rastreabilidade: "Plano de Projeto\r\n\
  Cronograma\r\n\
  Planilha de Custos\r\n\
  Planilha Tamanho de Projeto\r\n\
  Plano de Ger\xC3\xAAncia de Configura\xC3\xA7\xC3\xA3o\r\n\
  Lista de Riscos\r\n\
  Manual de Utiliza\xC3\xA7\xC3\xA3o"
analista_de_configuracao_id: c5832816e07274dc8b46799e19bf712a7e9c1ee1
id: 7459a32db026f1945c0e899558512e9f93b24f83
svn_projeto: ""
git_pasta: ""
created_at: 2012-11-27 07:54:23.026559 -02:00
auditor_de_ppqa_id: ""
```

**Quadro 3 – Exemplo 2 de arquivo YAML do banco de dados legado para entidade Projeto
(continuação)**

```

---
ativo: false
analista_de_ppqa_id: 670c924d035aa89186b290e6e3127a5055738fcc
user_ids:
- 857fcd88953583704d2c9f75cc1d2845d4e5716b
- 2a3fbb694588cc735c2b4f4832907a0c48e2ced3
- 670c924d035aa89186b290e6e3127a5055738fcc
- c5832816e07274dc8b46799e19bf712a7e9c1ee1
- 06ea31ca93cae8822ed7c832b6b5ca54f922c324
- 82f11a20f35e8d19b43893fceb0490ecea84d
- 8f9edb773957339c2288df4b6bc9cb898071e4eb
- d7c6f99d5e30f3110231f95994b627f2774381e5
- 5b5cdab5eec069d8e938c463119720210d602a71
- c59e249dc67eb3507a2397b9dd263b2ce5788676
fornecedores_de_requisitos: ""
gerente_de_projetos_id: c5832816e07274dc8b46799e19bf712a7e9c1ee1
data_prevista_de_conclusao:
fase_atual: ""
svn_projeto: ""
analista_de_configuracao_id: d7c6f99d5e30f3110231f95994b627f2774381e5
cargos: "Gerente de Projetos\r\n\
Gerente Administrativo\r\n\
Programador Efetivo\r\n\
Programador Estagi\r\n\
Analista de PPQA\r\n\
Auditor de PPQA\r\n\
Analista de CM\r\n\
Auditor de CM\r\n\
Designer\r\n\
Ilustrador\r\n\
Product Owner"
svn_produto: ""
auditor_de_configuracao_id: 857fcd88953583704d2c9f75cc1d2845d4e5716b
qtd_requisitos: 17
documentos_rastreabilidade: "Plano de Projeto\r\n\
Cronograma\r\n\
Planilha de Custos\r\n\
Planilha Tamanho de Projeto\r\n\
Plano de Ger\r\n\
Lista de Riscos\r\n\
Manual de Utiliza\r\n\
updated_at: 2013-04-02 20:33:43.804423 -03:00
id: 2ac9ed078f15138a9f7567704fa01004a7a9d620
created_at: 2013-01-09 16:58:11.168794 -02:00
auditor_de_ppqa_id: 8f9edb773957339c2288df4b6bc9cb898071e4eb
nome: Site
descricao: ""
metodos:
- 2c85003b97675a267d84a3c74f86471809cba1a9
- 7ab9b3fab900e1823eb66a6f8518dffae524a729
- ba1ce73ad4c279f6d15da54ba2555f59a911b0bf
- commit 6ab0c242bef82fc45b13508e6cb552b97e93b16e

```

Quadro 4 – Exemplo 3 de arquivo YAML do banco de dados legado para entidade Projeto

```

- 414aade75c331f7f9bab3d8203f0aaee885a6727
- aa158222d031b2e400f1f4462864b8e231e2b647
- 3ab70f39b482a3b09e49d58b9838ebd16362530d
- 241fc9547cb363516c12f339436862d300af47ed
- 4722150e03b3418feb243291d620cf81a309657b
- 84c4bb88021e011e1878b5154777a99baf2b4e77
- 1aa64911b964b1d54470803a0ebbd2f1d7fae91
- 7a20ff9dd0a9f47a04a4065bbce227cd645911d4
- 598f84b637d0a08a8336c9747c7eb7245412ea7f
- c83b1167f638e886bf28c5f66b1c13efaa11d5a3
- cce151bb715be5fce22e29fa9569ccb7a4ed28c1
- 1e63e05e88b3786fdca277a9de9548566b7742ba
- .
- 351c523c85cb80780df9b9ca940a23021f184716
- 774bf09ec2fd68458b0edb2ec59c8cca4c5d9f77
- e73d67ab3d2f50c994f429bd17f70151fc6da668
- issues
- e41df328913c6e01b716352ce47ae882d2a2c51d
git_pasta: ""

```

**Quadro 4 – Exemplo 3 de arquivo YAML do banco de dados legado para entidade Projeto
(continuação)**

5.1.2 Entidade Carta

A entidade “Carta”, atuando como registro de atividade ou de implementação de um requisito, foi armazenada em arquivos nomeados “card.yml”, no diretório reservado aos dados exclusivos do projeto, com atributos variáveis como apresentados nos Quadros 5, 6, 7 e 8. O nome da entidade “Carta” segue a nomenclatura proposta pelo método de estimativa *Extreme Programming* (XP), utilizado na época de desenvolvimento do software de planejamento e monitoramento de execução de atividades de desenvolvimento de software.

```

---
updated_at: &id001 2011-03-19 15:13:30.645097 -02:00
created_at: *id001
id: ba76d991d67ef93986ffc9d6f3eacebae6b963648
horas_contam_para_planejamento_do_projeto: true
tipo: tarefa_desenvolvimento
status: nao_iniciado
tarefa_id: 5ea495c863a74c39695dcc6a0b0d279889f36e44
project_id: 8f834fd27e5f0e42b4a0fbea590f2d5292b8acb3
prioridade: 1
titulo: Compor Requisitos

```

Quadro 5 – Exemplo 1 de arquivo YAML do banco de dados legado para entidade Carta

```

---
updated_at: &id001 2011-01-31 08:13:45.049020 -02:00
carta_de_desenvolvimento: false
tipo: historia
descricao: |-
  Banner Fixo;
  Imagens Fixas;
  Selecionar Eventos Randomicos;
status: nao_iniciado
esforco: 4.0
prioridade: 1000000
responsavel_id: c5832816e07274dc8b46799e19bf712a7e9c1ee1
horas_contam_para_planejamento_do_projeto: false
created_at: *id001
project_id: 76903e72697cdc63a66e930f1ae12d3b6864d312
cargo: ""
requisite_id: ""
id: 282814a9d2330ee9e8a39e1fc8c6c6017c152f9a
titulo: Layout Home
tarefa_id: d8386858356d0af87d011de746f33c5c7386bb9a
permite_execucao_em_grupo: false

```

Quadro 6 – Exemplo 2 de arquivo YAML do banco de dados legado para entidade Carta

```

---
carta_de_desenvolvimento: false
status: finalizado
project_id: 0ee8ac3a47b67ac33ca845f5f7e9c880f4c5707d
works:
- finalizado_em: 2011-04-04 15:14:04.291049 -03:00
  tempo_de_execucao: 2143.317355
  iniciado_em: 2011-04-04 14:38:20.973694 -03:00
  descricao: ok
  user_id: 1547efaabf881e837579449b08ff7f477b72ef5b
  metodos:
- finalizado_em: 2011-04-04 15:22:16.359144 -03:00
  tempo_de_execucao: 2534.51979
  descricao: Finalizada
  iniciado_em: 2011-04-04 14:40:01.839354 -03:00
  metodos:
  user_id: 4ab8eb9428933d1f7f8ea61d35ec753b5db12beb
- finalizado_em: 2013-04-04 15:09:04.182227 -03:00
  tempo_de_execucao: 1660.942386
  descricao: "reuni\&#xA3o validada."
  iniciado_em: 2011-04-04 14:41:23.239841 -03:00
  metodos:
  user_id: 06ea31ca93cae8822ed7c832b6b5ca54f922c324
requisite_id: ""
cargo: Gerente de Projetos
data_para_termino: 2013-04-04
data_para_inicio: 2013-04-04
permite_execucao_em_grupo: true
id: 8fea3ee2b31010725d865d2e268831cd25461656
updated_at: 2011-04-04 15:22:16.362922 -03:00
horas_contam_para_planejamento_do_projeto: true
responsavel_id: 1547efaabf881e837579449b08ff7f477b72ef5b
created_at: 2011-04-02 21:03:07.362780 -03:00
descricao: O gestor do projeto vai apresentar o projeto para o
desenvolvedor.
tipo: tarefa_desenvolvimento
tarefa_id: 0e1b9acc017bee06120103d3efb8fa9d71eef330
titulo: "Reuni\&#xA3o de Apresenta\&#xA7\&#xA3o do Projeto"
esforco: 3.0
prioridade: 1000000

```

Quadro 7 – Exemplo 3 de arquivo YAML do banco de dados legado para entidade Carta

```

---
carta_de_desenvolvimento: false
updated_at: 2011-01-28 09:08:00.433494 -02:00
descricao: ""
tipo: tarefa_desenvolvimento
esforco: 1.0
status: finalizado
works:
- !map:HashWithIndifferentAccess
  user_id: 06ea31ca93cae8822ed7c832b6b5ca54f922c324
  metodos: []

  descricao: |
    ok

  finalizado_em: 2011-01-25 17:33:00 -02:00
  tempo_de_execucao: 2460.0
  iniciado_em: 2011-01-25 16:52:00 -02:00
prioridade: 1000000
horas_contam_para_planejamento_do_projeto: true
responsavel_id: 06ea31ca93cae8822ed7c832b6b5ca54f922c324
requisite_id: ""
cargo: ""
project_id: 8623591abe56b0457b42655134569feab2bde44d
created_at: 2011-01-25 16:52:45.122705 -02:00
titulo: Monitoramento das atividades 25/01
id: 007a4db5efe8ed1f5e7e28b300c784c227c9efde
tarefa_id: ""
permite_execucao_em_grupo: false

```

Quadro 8 – Exemplo 4 de arquivo YAML do banco de dados legado para entidade Carta

Para esta entidade foram lidos os campos “id”, para referência durante a importação, “titulo” e “descricao”, para servir de descrição da atividade para análise, “project_id”, para relacionamento com a entidade Projeto importada, “created_at”, para definir quando foi cadastrada a carta com sua estimativa, e “esforco”, horas estimadas para execução da atividade.

Os registros de execução da atividade foram organizados como listas no campo “works”. Para cada registro de execução de atividade foram lidos os campos “iniciado_em” e “finalizado_em” para calcular o tempo de execução. Foi desconsiderado o campo “tempo_de_execucao” por que os valores deste campo, em vários casos, foram encontrados indefinidos ou muito diferentes da soma das diferenças entre o momento de início e término das execuções.

Além destes campos, para calcular o momento de início da execução da

atividade relativa ao projeto, foi utilizada a fórmula (4). E para calcular quanto do projeto já havia sido entregue antes do início da execução desta atividade, foi utilizada a fórmula (5).

$$\frac{DREA_i - DIPEPAP}{DFUEUAP - DIPEPAP} \quad (4)$$

$$\frac{\sum_{AF \in AFP} \left\{ \begin{array}{l} DFUE_{AF} \neq NULL \wedge DFUE_{AF} < DREA_i \quad THU_{AF} \\ \text{senão} \quad 0 \end{array} \right\}}{THUP} \quad (5)$$

Na fórmula (4), $DREA_i$ representa a data de realização de estimação da atividade em questão, $DIPEPAP$ representa a data inicial da primeira execução da primeira atividade do projeto, e $DFUEUAP$ representa a data final da última execução da última atividade do projeto. Já na fórmula (5), AF representa um registro de atividade finalizada, AFP representa o conjunto de atividades finalizadas do projeto, $DFUE_{AF}$ representa a data final da última execução de uma determinada atividade finalizada, $DREA_i$ representa a data de realização de estimação da atividade em questão, THU_{AF} representa o total de horas utilizadas de uma determinada atividade finalizada, e $HUTP$ representa o total de horas utilizadas do projeto.

5.1.3 Seleção das Atividades e Projetos Migrados

Para evitar que as atividades utilizadas no experimento fossem de difícil compreensão, incompletas ou relacionadas a projetos interrompidos ou de curta duração, foi gerada uma planilha com os dados colhidos dos projetos e atividades para filtro manual, selecionando 17 atividades para estimativa, todas de um mesmo projeto, e outras 100 atividades de outros 10 projetos, para revisão de atividades.

Os registros selecionados foram convertidos do formato de planilha para formato "YAML" com os campos de cada registro respeitando as tabelas do banco de dados, sendo facilmente recarregados no banco de dados sempre que

necessário.

5.2 FERRAMENTA DE MIGRAÇÃO DE NOTAS DOS PARTICIPANTES

Para investigar uma possível correlação entre notas dos participantes em disciplinas de computação com outros resultados encontrados, foram salvos os históricos escolares em formato HTML e, através desta ferramenta, lidos e convertidos para CSV, com médias calculadas por participante. Após a revisão dos dados convertidos, estes foram inseridos no banco de dados para serem utilizados nas telas de busca de correlação.

5.3 FERRAMENTA DE RECUPERAÇÃO DE REQUISIÇÕES DO NAVEGADOR

Esta ferramenta foi desenvolvida como uma forma de assegurar a captação dos dados, lendo o log de requisições do navegador realizados pelos usuários. E em caso de falhas na resposta da ferramenta de coleta, causada por bugs, recuperar os dados inseridos pelo usuário.

A recuperação de requisições do navegador foi necessária por se manter um ambiente de desenvolvimento de alteração frequente e de falta de rigor nos testes realizados da ferramenta de coleta. E se mostrou simples por fazer uso de arquivos de log gerados pelo *framework* de desenvolvimento RubyOnRails utilizado.

5.3.1 Exemplos de Entradas de Log

Como podem ser vistos nos exemplos de arquivo de log, nos Quadros 9, 10, 11, 12, e 13, as entradas no log de requisição do navegador são escritas para serem facilmente lidas por pessoas. E por manterem o padrão em sua escrita, puderam ser lidos e processados por esta ferramenta.

```

Started GET "/participants/new" for 127.0.0.1 at 2014-08-11 19:41:44
-0300
Processing by ParticipantsController#new as HTML
  Rendered participants/_form.html.erb (3.4ms)
  Rendered participants/new.html.erb within layouts/application (4.3ms)
Completed 200 OK in 8ms (Views: 7.7ms | ActiveRecord: 0.0ms)

```

Quadro 9 – Exemplo de entrada de *log* de requisição GET com sucesso

```

Started POST "/participants" for 127.0.0.1 at 2014-08-11 20:07:02 -0300
Processing by ParticipantsController#create as HTML
  Parameters: {"utf8"=>"✓",
"authenticity_token"=>"6BaoZu23KEZl1DS/c5i0EYEWYCBdn/J0AlB1Vz6wYSU=",
"participant"=>{"name"=>"Carlos", "birthday"=>"12/03/1992",
"function"=>"Analista", "sex"=>"1"}, "commit"=>"Salvar"}
  ESC[1mESC[35m (0.1ms)ESC[0m BEGIN
  ESC[1mESC[36mSQL (0.7ms)ESC[0m ESC[1mINSERT INTO `participants`
(`birthday`, `created_at`, `function`, `name`, `sex`, `updated_at`)
VALUES ('1990-07-30', '2014-08-11 23:07:02', 'Analista', 'André Tissot',
1, '2014-08-11 23:07:02')ESC[0m
  ESC[1mESC[35m (4.9ms)ESC[0m COMMIT
Redirected to http://localhost:3000/participants/1
Completed 302 Found in 17ms (ActiveRecord: 6.9ms)

```

Quadro 10 – Exemplo de entrada de *log* de requisição POST com sucesso

```

Started PATCH "/participants/2" for 127.0.0.1 at 2014-08-11 21:39:44
-0300
Processing by ParticipantsController#update as HTML
  Parameters: {"utf8"=>"✓",
"authenticity_token"=>"Q8frev+VuIy2oA3eIs91Kh1kiRbN2kKfeiuUkxjvTxA=",
"participant"=>{"name"=>"João", "birthday"=>"20/10/2000",
"function"=>"Programador", "sex"=>"1"}, "commit"=>"Salvar", "id"=>"2"}
  ESC[1mESC[36mParticipant Load (0.1ms)ESC[0m ESC[1mSELECT
`participants`.* FROM `participants` WHERE `participants`.`deleted_at`
IS NULL AND `participants`.`id` = 2 LIMIT 1ESC[0m
  ESC[1mESC[35m (2.1ms)ESC[0m BEGIN
  ESC[1mESC[36mSQL (6.3ms)ESC[0m ESC[1mUPDATE `participants` SET
`birthday` = '2000-10-20', `updated_at` = '2014-08-12 00:39:44' WHERE
`participants`.`id` = 2ESC[0m
  ESC[1mESC[35m (6.5ms)ESC[0m COMMIT
Redirected to http://localhost:3000/participants/2
Completed 302 Found in 20ms (ActiveRecord: 14.9ms)

```

Quadro 11 – Exemplo de entrada de *log* de requisição POST com sucesso simulando PUT

```
Started GET "/participants" for 127.0.0.1 at 2014-08-17 15:13:26 -0300
Processing by ParticipantsController#index as HTML
Completed 404 Not Found in 1ms

ActiveRecord::RecordNotFound (Couldn't find Team without an ID):
  app/controllers/participants_controller.rb:84:in `set_team'

  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/_source.erb
(1.6ms)
  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/_trace.html.erb
(1.4ms)
  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/_request_and_resp
onse.html.erb (1.2ms)
  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/diagnostics.erb
within rescues/layout (24.9ms)
```

Quadro 12 – Exemplo de entrada de log de requisição GET cujo recurso não foi encontrado

```

Started GET "/participants" for 127.0.0.1 at 2014-08-17 15:16:59 -0300
Processing by ParticipantsController#index as HTML
  ESC[1mESC[35mTeam Load (0.2ms)ESC[0m SELECT `teams`.* FROM `teams`
WHERE `teams`.`deleted_at` IS NULL AND `teams`.`id` = 1 LIMIT 1
  ESC[1mESC[36mTeamParticipant Load (0.1ms)ESC[0m ESC[1mSELECT
`team_participants`.* FROM `team_participants` WHERE
`team_participants`.`deleted_at` IS NULL AND
`team_participants`.`team_id` = 1ESC[0m
  ESC[1mESC[35mParticipant Load (0.1ms)ESC[0m SELECT `participants`.*
FROM `participants` WHERE `participants`.`deleted_at` IS NULL
  Rendered participants/_header.html.erb (0.3ms)
  Rendered participants/_list.html.erb (4.0ms)
  Rendered participants/index.html.erb within layouts/application
(24.3ms)
Completed 500 Internal Server Error in 31ms

ActionView::Template::Error (undefined method `name' for nil:NilClass):
  2:   <table>
  3:     <thead>
  4:       <tr>
  5:         <th colspan="2">Integrantes do time <%= @time.name
%></th>
  6:       </tr>
  7:       <tr>
  8:         <th>Nome</th>
  app/views/participants/_list.html.erb:5:in
`_app_views_participants_list_html_erb__1432341903606958197_43597760'
  app/views/participants/index.html.erb:2:in
`_app_views_participants_index_html_erb__2012899283318212817_69939583979
840'

  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/_trace.html.erb
(1.7ms)
  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/_request_and_resp
onse.html.erb (1.0ms)
  Rendered /home/andre/.rvm/gems/ruby-2.1.2/gems/actionpack-
4.1.4/lib/action_dispatch/middleware/templates/rescues/template_error.ht
ml.erb within rescues/layout (17.2ms)

```

Quadro 13 – Exemplo de entrada de log de requisição GET com erros

5.3.2 Banco de Dados

A ferramenta de recuperação de requisições do navegador utiliza o sistema gerenciador de banco de dados (SGBD) MariaDB por ter um interpretador *Structured Query Language* (SQL) e ter seu código-fonte baseado no MySQL. Porém diferente

do MySQL, este tem como objetivo se manter completamente *open-source*. Por ser um SGBD SQL, se torna necessária a organização dos dados em tabelas, estas tabelas apresentadas na Figura 11.

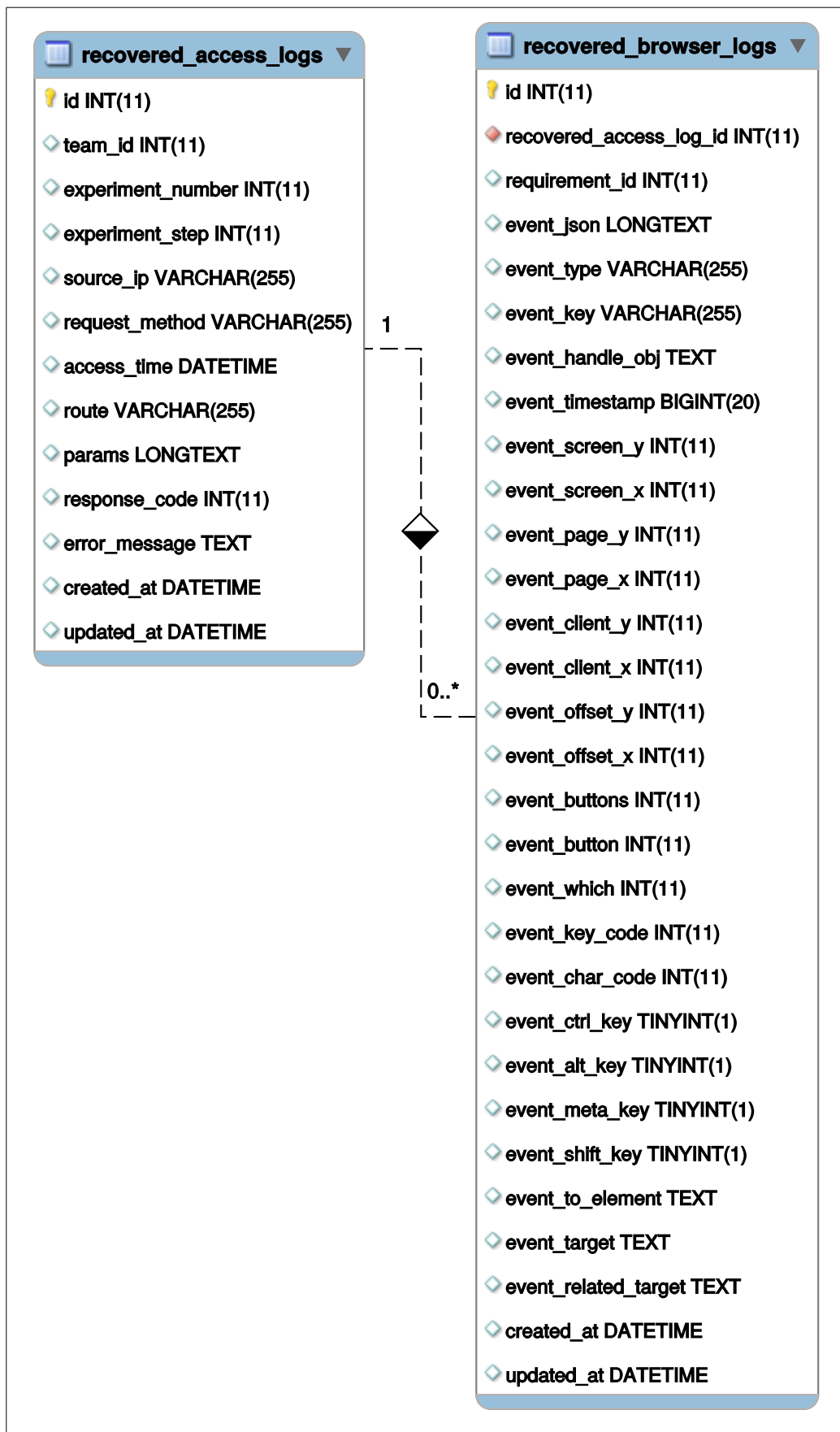


Figura 11 – Tabelas do banco de dados da ferramenta de recuperação de logs de requisição de navegador.

5.4 FERRAMENTA DE COLETA DE DADOS

A ferramenta web faz a captura de diversos eventos do navegador, para traçar o raciocínio dos participantes no processo de definição da estimativa, além dos dados inseridos pelos participantes.

5.4.1 Telas

As telas foram construídas baseadas nos rascunhos de tela apresentados no capítulo 4.6.3. Com o objetivo de evitar poluição visual, foi implementada uma interface simples, com elementos visuais em branco e cinza, texto comum em preto, com títulos em negrito, e a cor vermelha para mensagens de validação, como apresentado em mais detalhes neste capítulo.

5.4.1.1 Definição do Grupo de Estimativa

Ao acessar a ferramenta web de coleta de dados, os participantes visualizam os grupos de estimativa cadastrados, chamados na ferramenta de times, como na Figura 12. Esta tela ilustra a interface utilizada para a atividade “Participantes procuram o cadastro do seu grupo” do diagrama da Figura 3.

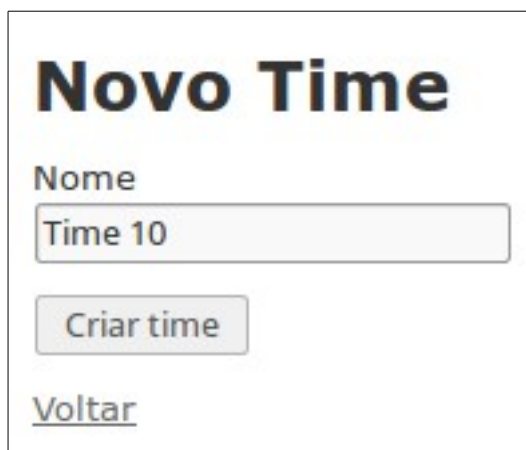


Nome	<u>Novo time</u>
Time 1	<u>Selecione</u>
Time 2	<u>Selecione</u>
Time 3	<u>Selecione</u>
Time 4	<u>Selecione</u>

Figura 12 – Captura da tela de listagem de grupos de estimativa cadastrados.

Caso os participantes encontrem o grupo de estimativa de interesse, podem clicar em “Selecione” ao lado do grupo desejado. Esta ação executa a atividade “Participantes selecionam seu grupo” do diagrama da Figura 3.

Caso os participantes não encontrem o grupo de estimativa de interesse e desejem cadastrá-lo, podem clicar em “Novo time” no canto superior direito da tabela de busca, para exibir a tela de cadastro de grupo de estimativa, como na Figura 13, e inserir o nome do seu grupo de análise. Esta ação executa a atividade “Participantes cadastram seu grupo” do diagrama da Figura 3.



A captura de tela mostra uma interface web com o título "Novo Time" em uma fonte grande e bold. Abaixo do título, há o rótulo "Nome" seguido por um campo de entrada de texto contendo o texto "Time 10". Logo abaixo do campo de texto, há um botão retangular com o texto "Criar time". Na base da interface, há um link "Voltar" sublinhado.

Figura 13 – Captura da tela de cadastro de grupo de estimativa.

5.4.1.2 Seleção e Cadastro dos Participantes

As telas apresentadas neste capítulo foram projetadas no capítulo 4.6.3.1 e implementadas, sendo alteradas de acordo com a necessidade.

Ao cadastrar um novo grupo de estimativa, pelo formulário apresentado na Figura 13, ou selecionar um grupo de estimativa da lista, na tabela apresentada na Figura 12, será exibida a tela de seleção e cadastro dos participantes com os participantes integrantes do grupo de estimativa, na tabela localizada à esquerda, e os demais participantes cadastrados, na tabela localizada à direita, como na Figura 14. Esta tela ilustra a interface utilizada para a atividade “Participantes verificam a lista de integrantes do grupo” do diagrama da Figura 4.

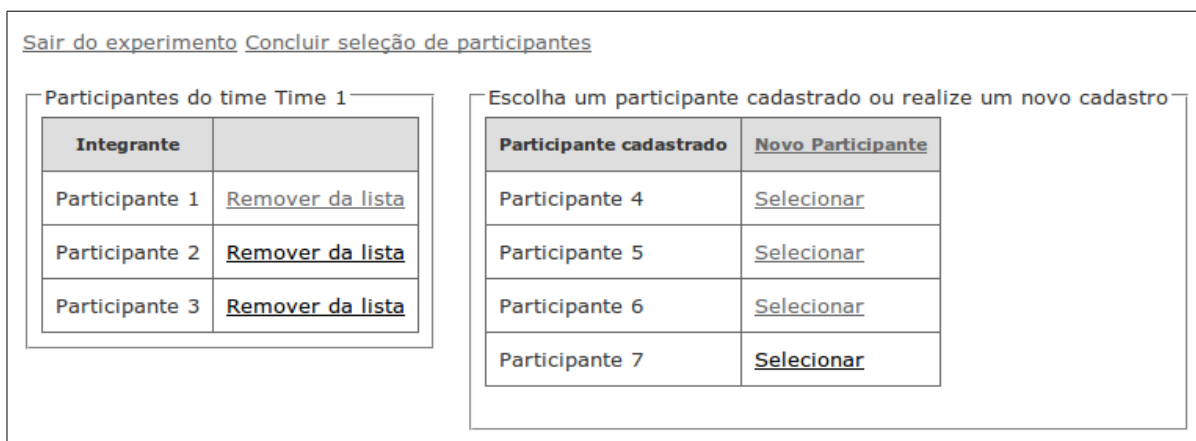


Figura 14 – Captura da tela inicial de seleção e cadastro de participantes.

Caso os participantes queiram retornar a seleção do grupo de estimativa, podem clicar em “Sair do experimento”, retornando a tela de seleção de grupos de estimativa, como na Figura 12.

Caso os participantes percebam que há um ou mais cadastros de participantes integrantes desse grupo que foram erroneamente relacionados a este grupo, podem remover este integrante do grupo ao clicar em “Remover da lista”. Esta tela ilustra a interface utilizada para a atividade “Participantes removem um dos participantes da lista de selecionados” do diagrama da Figura 4.

Caso os participantes percebam que há um ou mais participantes a adicionar ao grupo de estimativa, poderão procurar na tabela de participantes cadastrados pelo seu(s) nome(s). Esta ação executa a atividade “Participantes procuram o cadastro do participante” do diagrama da Figura 4.

Caso os participantes não encontrem o cadastro de um participante na listagem de participantes cadastrados e cliquem em “Novo Participante” no canto superior direito da tabela de participantes cadastrados, será exibida a tela de cadastro de participante integrante do grupo de estimativa, como na Figura 15, para inserirem os dados pessoais e competências profissionais e acadêmicas do participante e clicarem em “Salvar” para gravar os dados do participante como integrante do grupo de estimativa. Esta tela ilustra a interface utilizada para a atividade “Participantes cadastram os dados do participante” do diagrama da Figura 4.

[Sair do experimento](#) [Concluir seleção de participantes](#)

Participantes do time Time 1

Integrante	Mostrar lista de cadastrados
Participante 1	Remover da lista
Participante 2	Remover da lista
Participante 3	Remover da lista

Participante 4

Nome Completo:

Data de Nascimento:

Competência	Básico	Intermediário	Avançado	
<input type="text" value="Estimativa de Software"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Remover
<input type="text" value="Java"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Remover
<input type="text" value="Sistemas Operacionais"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Caso trabalhe, função que desempenha:

Sexo: Masculino Feminino

| [Voltar](#)

Figura 15 – Captura da tela de cadastro de participante.

Caso os participantes encontrem o cadastro do participante na listagem de participantes cadastrados e cliquem em “Selecionar” ao lado do cadastro escolhido, será exibida a tela de revisão dos dados cadastrados, como na Figura 16. Esta tela ilustra a interface utilizada para a atividade “Participantes revisam os dados do cadastro” do diagrama da Figura 4.

[Sair do experimento](#) [Concluir seleção de participantes](#)

Participantes do time Time 1

Integrante	Mostrar lista de cadastrados
Participante 1	Remover da lista
Participante 2	Remover da lista
Participante 3	Remover da lista

Tem certeza que deseja selecionar este participante?

Nome: Participante 4

Data de Nascimento: 12/12/1985

Competência	Nível
Estimativa de Software	Básico
Java	Básico
Sistemas Operacionais	Básico
Licença de software	Básico
C	Básico

Caso trabalhe, função que desempenha: Analista de Sistemas

Sexo: Masculino

[Selecionar](#) | [Editar](#) | [Voltar](#)

Figura 16 – Captura da tela de revisão do cadastro de participante.

Caso os participantes não concordem com os dados exibidos na tela apresentada no Figura 16 e cliquem em “Editar”, será exibida a tela com os dados cadastrados para alteração, como na Figura 17. Esta tela ilustra a interface utilizada para a atividade “Participantes alteram os dados cadastrados” do diagrama da Figura 4.

Sair do experimento [Concluir seleção de participantes](#)

Participantes do time Time 1

Integrante	Mostrar lista de cadastrados
Participante 1	Remover da lista
Participante 2	Remover da lista
Participante 3	Remover da lista

Participante 4

Nome Completo:

Data de Nascimento:

Competência	Básico	Intermediário	Avançado	
<input type="text" value="Estimativa de Software"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text" value="Java"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text" value="Sistemas Operacionais"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text" value="Licença de software"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text" value="C"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remover
<input type="text"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	

Caso trabalhe, função que desempenha:

Sexo: Masculino Feminino

| [Voltar](#)

Figura 17 – Captura da tela de alteração cadastro de participante.

5.4.1.3 Definição de Estimativas

As telas apresentadas neste capítulo foram projetadas nas seções 4.6.3.2 e 4.6.3.3 e implementadas, sofrendo alterações inerentes à revisita dos autores aos rascunhos de tela.

Caso os participantes decidam encerrar a seleção de participantes e cliquem em “Concluir seleção de participantes”, será exibida a tela de listagem de requisitos a estimar, como na Figura 18. Esta tela ilustra a interface utilizada para a atividade “Participantes acessam a lista de requisitos a estimar” do diagrama da Figura 4.

Time: "Time 1" Sair do experimento

Requerimentos para estimativa

Requisito 1	<u>Abrir</u>
Requisito 2	<u>Abrir</u>
Requisito 3	<u>Abrir</u>
Requisito 4	<u>Abrir</u>
Requisito 5	<u>Abrir</u>
Requisito 6	<u>Abrir</u>

Figura 18 – Captura da tela de listagem de requisitos a estimar.

Ao clicar em “Abrir” em uma das opções da lista de requisitos, será exibida a tela e exibição dos detalhes do requisito a estimar selecionado, como na Figura 19. A área de detalhes do requisito desta tela ilustra a componente gráfico utilizado para a atividade “Participantes leem os dados referentes ao requisito” do diagrama da Figura 5.

Time: "Time 1" [Sair do experimento](#)

Requerimentos para estimativa

Requisito 1	Abrir
Requisito 2	Abrir
Requisito 3	Abrir
Requisito 4	Abrir
Requisito 5	Abrir
Requisito 6	Abrir

Requisito 1

Objetivos do projeto: Criação de ferramenta para gerenciamento de eventos.

Tecnologias empregadas: PHP, MySQL, Javascript.

Tamanho estimado do projeto: 400 homens-hora.

Momento do projeto: 40%

Quanto já foi realizado: 50%

Descrição:
Adicionar área de "fale conosco" com os campos nome, e-mail e mensagem. E ao cadastrar esta mensagem no banco de dados, deve ser enviado um e-mail para administrador@sistema.com.br. Uma tabela no banco de dados deverá ser criada para armazenar as mensagens cadastradas.

Rodada 2

Nome	Valor Anterior	Valor Atual	Valor Justificativa
Participante 1	4.0	<input type="text" value="3"/>	...
Participante 2	6.0	<input type="text" value="4"/>	...
Participante 3	8.0	<input type="text" value="6"/>	...

| [Definir estimativa final](#)

Figura 19 – Captura da tela de exibição do requisito a estimar e de execução das rodadas de *Planning Poker*.

Na ferramenta web, ao preencherem os campos de valor de estimativa e de justificativa para cada estimador, exibidos na Figura 19, e clicarem em “Terminar rodada”, os dados respectivos a esta rodada serão gravados, e a tela é atualizada para permitir uma nova rodada, exibindo os valores anteriores estimados como “Valor Anterior” para que o moderador possa saber se os estimadores estão mudando suas estimativas ou não estão chegando a um consenso. Esta ação executa a atividade “Participantes realizam as rodadas de jogo, definindo e justificando o esforço estimado” do diagrama da Figura 5.

Caso os participantes cheguem a um consenso ou o moderador decida encerrar as rodadas por falta de consenso, decidam definir o valor de estimativa final e cliquem em “Definir estimativa final”, a tela de estimativa final é exibida, como na Figura 20.

Time: "Time 1" [Sair do experimento](#)

Requerimentos para estimativa	
Requisito 1	Abrir
Requisito 2	Abrir
Requisito 3	Abrir
Requisito 4	Abrir
Requisito 5	Abrir
Requisito 6	Abrir

Requisito 1

Objetivos do projeto: Criação de ferramenta para gerenciamento de eventos.

Tecnologias empregadas: PHP, MySQL, Javascript.

Tamanho estimado do projeto: 400 homens-hora.

Momento do projeto: 40%

Quanto já foi realizado: 50%

Descrição:
Adicionar área de "fale conosco" com os campos nome, e-mail e mensagem. E ao cadastrar esta mensagem no banco de dados, deve ser enviado um e-mail para administrador@sistema.com.br. Uma tabela no banco de dados deverá ser criada para armazenar as mensagens cadastradas.

Estimativa final:

Descreva livremente, e com o máximo de detalhes as razões e incertezas que o(s) levou(aram) a esta estimativa:

Criar tabelas em SQL é rápido, mas testar operações de envio de e-mail leva mais tempo.

Figura 20 – Captura da tela de exibição do requisito a estimar e de definição da estimativa final.

Ao preencherem os campos de valor de estimativa final e de justificativa, exibidos na Figura 20, e clicarem “Salvar”, estes dados serão gravados e essa será reexibida, porém sem campos de entrada de dados, como na Figura 21. Esta ação executa a atividade “Participantes definem a estimativa final e a justificam” do diagrama da Figura 5.

Time: "Time 1" [Sair do experimento](#)

Requerimentos para estimativa	
Requisito 1	Abrir
Requisito 2	Abrir
Requisito 3	Abrir
Requisito 4	Abrir
Requisito 5	Abrir
Requisito 6	Abrir

Requisito 1

Objetivos do projeto: Criação de ferramenta para gerenciamento de eventos.

Tecnologias empregadas: PHP, MySQL, Javascript.

Tamanho estimado do projeto: 400 homens-hora.

Momento do projeto: 40%

Quanto já foi realizado: 50%

Descrição:
Adicionar área de "fale conosco" com os campos nome, e-mail e mensagem. E ao cadastrar esta mensagem no banco de dados, deve ser enviado um e-mail para administrador@sistema.com.br. Uma tabela no banco de dados deverá ser criada para armazenar as mensagens cadastradas.

Figura 21 – Captura da tela de exibição do requisito estimado.

Após estimarem todos os requisitos ou terem de encerrar o experimento, os participantes deverão clicar em “Sair do experimento”, retornando assim a tela de seleção de grupos de estimativa, como na Figura 12.

5.4.1.4 Repositório de Atividades Executadas

As telas apresentadas neste capítulo foram projetadas no capítulo 5.4.1.4 e implementadas, sofrendo alterações inerentes à revisita dos autores ao rascunho de tela.

Para verificar o impacto da utilização da busca em repositório durante a execução de estimativas utilizando *Planning Poker*, estas telas e suas funcionalidades estiveram apenas disponíveis na segunda etapa do experimento, relacionando assim as mudanças dos resultados ao uso deste recurso.

Ao acessarem este recurso diretamente pelo navegador, os participantes acessam a tela de busca de atividades, como na Figura 22.

Figura 22 – Captura da tela de busca de atividades executadas.

Ao preencherem os filtros de palavras no campo descrição, por projeto e/ou por esforço executado mínimo e máximo e clicarem em “Buscar”, são exibidas abaixo da área de busca, uma tabela com dados das atividades de desenvolvimento de software encontradas, como na Figura 23. Esta ação executa a atividade “Participantes buscam atividades já executadas no banco de dados” do diagrama da Figura 5.

Projeto (homens-hora)	Momento de Estimativa	Esforço Realizado	Descrição	
Projeto 03 (411 homens-hora)	97.16%	1.74 homens-hora	Manter Perfis de Acesso ----- Esta carta compreende: -Criação...	Abrir
Projeto 03 (411 homens-hora)	97.16%	6.14 homens-hora	Manter Usuários ----- Esta carta compreende: -A criação da ta...	Abrir
Projeto 03 (411 homens-hora)	97.18%	3.22 homens-hora	Manter Colegiados ----- Esta carta compreende: * Criação da t...	Abrir
Projeto 03 (411 homens-hora)	98.9%	1.61 homens-hora	Manter Atas ----- Esta carta compreende: * Criação da tabela ...	Abrir
Projeto 03 (411 homens-hora)	98.9%	1.15 homens-hora	Manter Grupos de Documentos ----- Esta carta compreende: * Cr...	Abrir
Projeto 03 (411 homens-hora)	98.93%	1.52 homens-hora	Manter Banners ----- Esta carta compreende: * Criação da tabe...	Abrir

Figura 23 – Captura da tela de busca de atividades executadas com os resultados encontrados.

Ao clicarem em “Abrir” ao lado da atividade escolhida, os dados desta são carregados abaixo da tabela de resultados, como na Figura 24, com uma configuração de informações similar a configuração das telas de estimativa, porém com o campo de “Esforço realizado” com o esforço executado na implementação da atividade. Esta ação executa a atividade “Participantes leem os dados referentes à atividade já executada” do diagrama da Figura 5.

Filtros de Busca

Descrição:

Projeto:

Esforço realizado mínimo:

Esforço realizado máximo:

Resultados da Busca (6 resultados)

Projeto (homens-hora)	Momento de Estimativa	Esforço Realizado	Descrição	
Projeto 03 (411 homens-hora)	97.16%	1.74 homens-hora	Manter Perfis de Acesso ----- Esta carta compreende: -Criação...	Abrir
Projeto 03 (411 homens-hora)	97.16%	6.14 homens-hora	Manter Usuários ----- Esta carta compreende: -A criação da ta...	Abrir
Projeto 03 (411 homens-hora)	97.18%	3.22 homens-hora	Manter Colegiados ----- Esta carta compreende: * Criação da t...	Abrir
Projeto 03 (411 homens-hora)	98.9%	1.61 homens-hora	Manter Atas ----- Esta carta compreende: * Criação da tabela ...	Abrir
Projeto 03 (411 homens-hora)	98.9%	1.15 homens-hora	Manter Grupos de Documentos ----- Esta carta compreende: * Cr...	Abrir
Projeto 03 (411 homens-hora)	98.93%	1.52 homens-hora	Manter Banners ----- Esta carta compreende: * Criação da tabe...	Abrir

Dados do Requisito

Objetivos do projeto: Criação de ferramenta para gerenciamento de eventos.

Tecnologias empregadas: PHP, MySQL, Javascript.

Tamanho estimado do projeto: 411 homens-hora.

Momento do projeto: 98.9%.

Quanto já foi realizado: 78.47%.

Esforço realizado: 1.15 homens-hora.

Descrição:
Manter Grupos de Documentos

Esta carta compreende:

- * Criação da tabela de document_groups se não já não estiver;
- * Criação das telas de CRUD para grupos de documentos.

Um grupo de documento possui os seguintes atributos:

- * name (string);
- * idiom (string);

Estas atividades devem ser realizadas dentro da área de administração.

Figura 24 – Captura da tela de busca de atividades executadas com detalhes da atividade executada.

5.4.2 Registro de Eventos

Além de serem registradas todas as requisições do navegador na forma do

arquivo de log, apresentado no capítulo 5.3, também são registrados os acessos às páginas, incluindo informações temporárias de sessão, e eventos de navegador, como cliques do mouse e escrita utilizando o teclado. Este registro de ações permitem verificar o comportamento do usuário, como, por exemplo, se este utilizava mais o mouse ou a tecla de tabulação, quanto tempo levava para escrever em cada campo, se apagava o texto que escrevia, se escrevia com cuidado pausadamente ou pensava bem antes de escrever ou se respondia rapidamente, se pulava atividades para estimá-las depois ou respondia assim que a visualizava, permitindo em trabalhos futuros relacionar este comportamento com o resultado do grupo de estimadores.

5.4.3 Banco de Dados

A ferramenta de coleta utiliza o sistema gerenciador de banco de dados (SGBD) MariaDB. E por ser um SGBD SQL, se torna necessária a organização dos dados em tabelas, estas tabelas apresentadas na Figura 25.

Para obter *logs* de acessos a ferramenta, com um objetivo parecido com o da ferramenta de recuperação de requisições do navegador, porém gravando também dados de estado da ferramenta de coleta como as de sessão do usuário logado, foram armazenados dados de acesso nas tabelas apresentadas na Figura 26. E após a evolução da ferramenta, na segunda iteração, as tabelas para este fim passaram a ser como apresentadas na Figura 27.

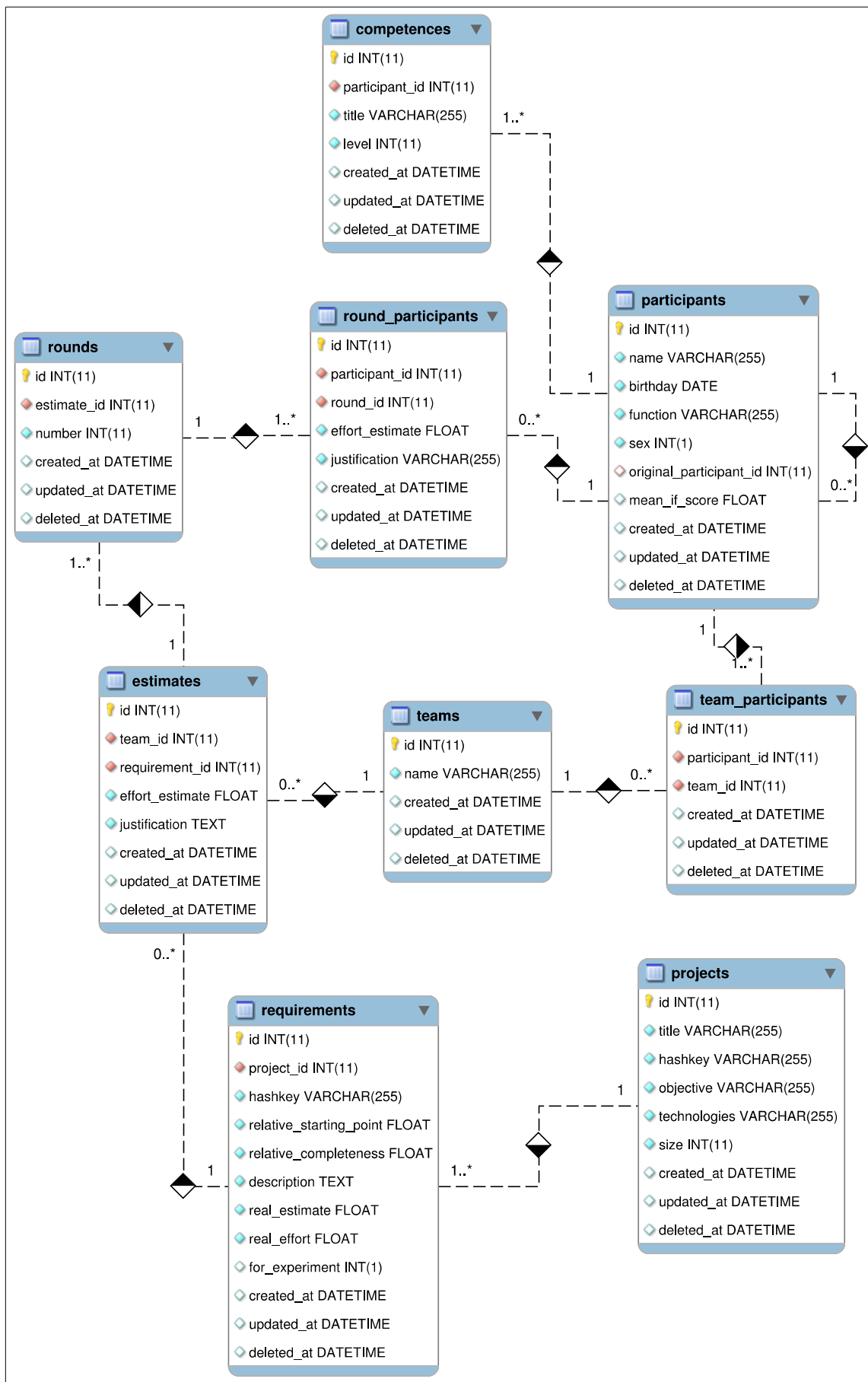


Figura 25 – Tabelas do banco de dados da ferramenta de coleta.

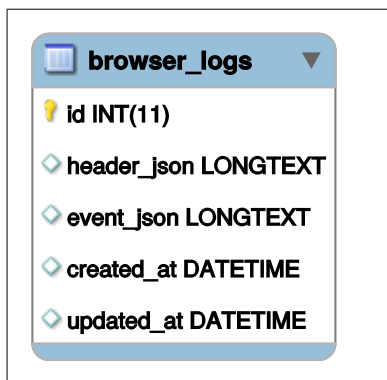


Figura 26 – Tabela do banco de dados da ferramenta de coleta para *log* interno de acesso pelo navegador na primeira iteração.

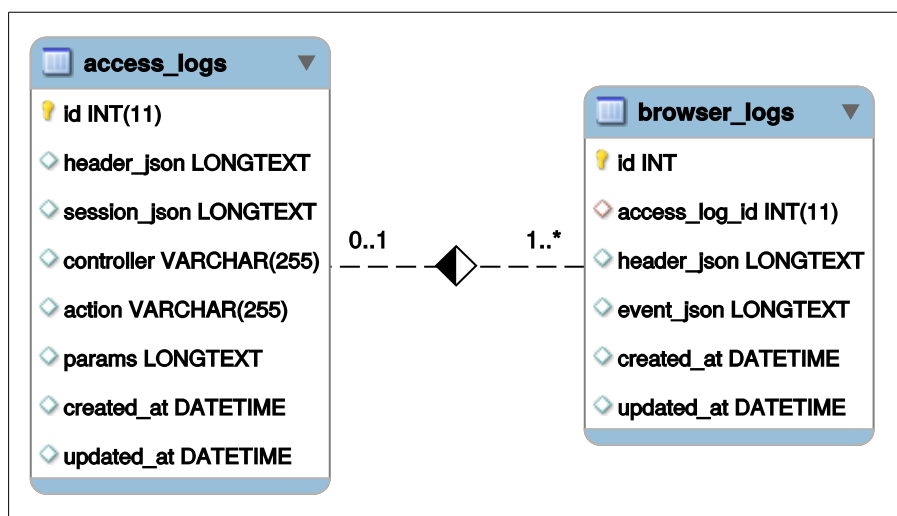


Figura 27 – Tabela do banco de dados da ferramenta de coleta para *log* interno de acesso pelo navegador na segunda iteração.

5.4.4 Alterações Após Primeira Iteração

Após a aplicação da ferramenta de coleta, foram notados alguns problemas ao observar o uso desta pelos participantes. Estes foram corrigidos com ações descritas nesse capítulo.

5.4.4.1 Adicionar Validação para Evitar que os Usuários Insiram Estimativas sem Realizar as Rodadas

Na versão original da ferramenta a tela de inserção de dados de estimativa, Figura 19, era possível clicar no link “definir estimativa final”, para exibir a tela da Figura 20, sem enviar os valores inseridos nos campos “Valor atual” e “Valor justificativa” da rodada de estimativa. O que fazia com que o servidor não recebesse os dados da rodada final, e apenas o valor e justificativa finais. Este defeito foi resolvido ao manter uma validação que apenas permite a troca das telas citadas se os valores dos campos de rodada estiverem vazios.

5.4.4.2 Separar Dados de Atividade e Dados de Projeto

Foi notada a confusão dos participantes a diferenciar os dados do projeto e do requisito, e por conta disto e de estimar as atividades de desenvolvimento de um único projeto, foram separadas estas informações, como na Figura 28, substituindo a tela apresentada na Figura 19.

Time: "teste" [Banco de dados histórico](#) [Sair do experimento](#)

Projeto

Objetivos do projeto: Portal de transparência para gastos municipais.

Tecnologias empregadas: PHP 5.4, MySQL 5.3, HTML 4, CSS 2, jQuery 1.8.

Tamanho estimado do projeto: 1170 homens-hora.

Requisitos a estimar

Requisito 1	Abrir
Requisito 2	Abrir
Requisito 3	Abrir
Requisito 4	Abrir
Requisito 5	Abrir
Requisito 6	Abrir
Requisito 7	Abrir
Requisito 8	Abrir
Requisito 9	Abrir
Requisito 10	Abrir
Requisito 11	Abrir
Requisito 12	Abrir
Requisito 13	Abrir
Requisito 14	Abrir
Requisito 15	Abrir
Requisito 16	Abrir
Requisito 17	Abrir

Requisito 1

Quanto tempo do cronograma total do projeto foi gasto antes do início da execução desta atividade: 80.82%.

Quanto do projeto já foi entregue antes do início da execução desta atividade: 63.59%.

Descrição:
Manter UploadViagens

Na Area administrativas criar o CRUD de UploadViagens.

A index deve listar os dados por ano/mes em ordem decrescente e paginado.

No formulario de Criação/Edição, permitir fazer upload de um arquivo. Salvar o endereço no campo arquivo. Na tela de editar, o sistema não deve apagar o arquivo já enviado se o não for selecionado um arquivo novo.

Todos os Campos são obrigatorios. (Cuidado, na tela de edição, o sistema n deve obrigar o usuario a subir novamente o mesmo aquivo já enviado, e nem apagar o arquivo já enviado se um novo n for selecionado).

A exclusão será fisica.

Rodada 1

Nome	Valor Anterior	Valor Atual	Valor Justificativa
teste		<input type="text"/>	<input type="text"/>

Figura 28 – Captura da tela aprimorada de exibição do requisito estimado.

5.4.4.3 Apenas Exibir o *Link* “definir valor final” se já Houver ao menos uma Rodada Executada

Antes os participantes eram capazes de acessar a tela de definição de estimativa final, tela da Figura 20, sem realizar nenhuma rodada do *Planning Poker*, na tela da Figura 28. Para corrigir este defeito, foi mantido escondido o link “definir valor final” enquanto não for realizada nenhuma rodada do *Planning Poker* para o requisito selecionado.

5.4.4.4 Mudar as *Labels* de Valores Relativos ao Momento e ao Conteúdo Entregue no Projeto para Ficarem Mais Claros.

Foi notada a confusão dos participantes na compreensão do significado dos dados de requisito “Momento do projeto” e “Quanto já foi realizado”, e por isso foram substituídos por “Quanto tempo do cronograma total do projeto foi gasto antes do início da execução desta atividade” e “Quanto do projeto já foi entregue antes do início da execução desta atividade”, respectivamente, como exibido na Figura 28.

5.4.4.5 Colocar o *Link* dos Dados Históricos Após a Seleção do Grupo de Estimadores para Manter *Link* com o Grupo

Ao realizar a análise inicial dos dados coletados, foi visto que vários participantes utilizavam outros computadores para acessar a base de dados históricos. E como este era acessado diretamente sem passar pela tela de seleção de time, o comportamento de busca de atividades executadas não pode ser relacionado ao time.

A alteração realizada foi criar um link na tela de realização de estimativas, como exibido na Figura 28, e bloquear o acesso direto por endereço. Deste modo, obrigando os participantes que pesquisam a identificarem o time ao qual pertencem.

5.4.4.6 Não Exibir na Busca os Projetos sem Requisitos da Busca

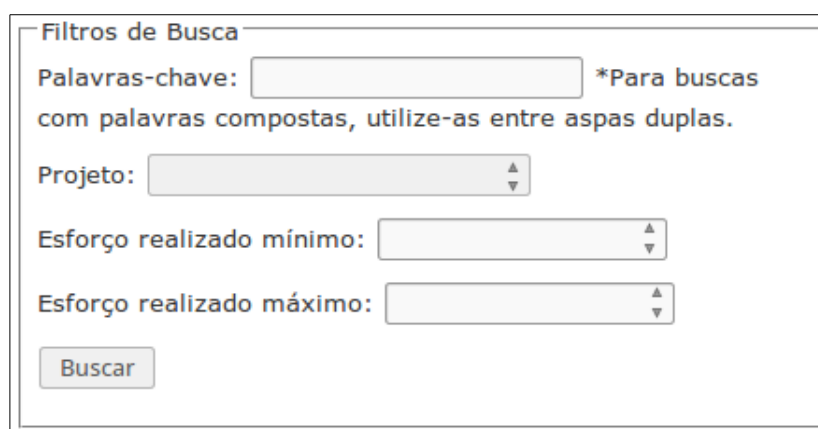
No filtro “Projeto” de busca da tela de busca de atividades executadas, exibido na Figura 22, foi removido o primeiro projeto da caixa de seleção, por este ter todos os requisitos sendo usados apenas para realização de estimativa.

5.4.4.7 Na Busca por Palavra Chave, Buscar Também em Tecnologias e Objetivos dos Projetos

Para auxiliar na busca de atividades executadas, os textos inseridos no campo de filtro “Descrição”, passaram a ser buscados também no campo “Tecnologia empregada”.

5.4.4.8 Alterar *Label* na Busca por “Palavras-chave”

Foi alterado o nome do campo do filtro de busca de atividades executadas de “Descrição” para “Palavras-chave”, como na Figura 26, para tornar mais claro que seu conteúdo não precisa ser igual ao da descrição completa do requisito.



A imagem mostra um formulário de busca com o título "Filtros de Busca". O formulário contém os seguintes elementos:

- Um campo de texto rotulado "Palavras-chave:" com um ícone de lupa à esquerda e o texto "*Para buscas com palavras compostas, utilize-as entre aspas duplas." à direita.
- Um menu suspenso rotulado "Projeto:" com uma seta para cima e uma seta para baixo.
- Um menu suspenso rotulado "Esforço realizado mínimo:" com uma seta para cima e uma seta para baixo.
- Um menu suspenso rotulado "Esforço realizado máximo:" com uma seta para cima e uma seta para baixo.
- Um botão rotulado "Buscar" na parte inferior esquerda.

Figura 29 – Captura da tela aprimorada de busca de atividades executadas.

5.4.4.9 Na Busca por Palavra Chave, Usar as Várias Palavras como Palavras-Chave Separadas e Ordenar os Resultados Contando as Palavras-Chave Encontradas

Para auxiliar os participantes a encontrarem resultados mais pertinentes, as

palavras inseridas no campo “Palavras-chave” do filtro de busca de atividades executadas, Figura 29, passaram a ser pesquisadas individualmente e em conjunto, ao escrever entre aspas duplas. Por exemplo ““PHP 5”” será visto como uma palavra, já o texto ‘nova tabela’, será visto como duas palavras-chave, por não estarem entre aspas duplas. Para deixar mais claro este aspecto do mecanismo de busca, foi adicionado o texto “*Para buscas com palavras compostas, utilize-as entre aspas duplas.” para informar ao participante desta função de busca, como exibido na Figura 29.

Os resultados da pesquisa de atividades executadas passaram a ser ordenadas, da atividade com o maior número de ocorrências das palavras-chave em sua descrição e/ou tecnologia empregada a com o menor número de ocorrências.

5.4.4.10 Adicionar Número de Resultados Encontrados na Busca

Para auxiliar o participante na decisão de filtrar melhor os resultados ou aceitar os que foram encontrados, foi adicionado o número de resultados da busca, como na Figura 30.

Filtros de Busca

Palavras-chave: *Para buscas com palavras compostas, utilize-as entre aspas duplas.

Projeto:

Esforço realizado mínimo:

Esforço realizado máximo:

Resultados da Busca (100 resultados)

Projeto (homens-hora)	Momento de Estimativa	Esforço Realizado	Descrição	
Projeto 02 (749)	0.02%	14.98 homens-hora	Agendamento de Questões - Parte I ----- Criar a tabela agenda...	Abrir
Projeto 02 (749)	7.49%	0.49 homens-hora	Matrícula Parte II ----- Na interface \"show\" de turma dever...	Abrir
Projeto 02 (749)	7.49%	8.78 homens-hora	Matrícula - Parte I ----- Deverá ser criado a tabela matricul...	Abrir
Projeto 02 (749)	56.75%	1.07 homens-hora	Layout Logos ----- Na logo no topo da pagina, o colocar margi...	Abrir
Projeto 02 (749)	56.75%	0.69 homens-hora	Ordenar Questões ----- Na listagem de questões do Aluno, orde...	Abrir
Projeto 02 (749)	60.3%	0.93 homens-hora	Layout Mobile ----- No layout Mobile colocar um div na parte ...	Abrir
Projeto 02 (749)	63.55%	8.14 homens-hora	Manter Horários de Envio ----- Manter Horarios. Cada contrat...	Abrir
Projeto 02 (749)	65.24%	0.8 homens-hora	Validação solicitar Qustão - Parte 001/3 ----- Criar campo na...	Abrir
Projeto 03 (411)	97.16%	0.85 homens-hora	Ativar auditoria automática ----- Criar tabela de auditoria e...	Abrir
Projeto 03 (411)	97.16%	0.98 homens-hora	Criar namespace de Administração ----- As páginas dentro de d...	Abrir
Projeto 03 (411)	97.16%	1.74 homens-hora	Manter Perfis de Acesso ----- Esta carta compreende: -Criação...	Abrir

Figura 30 – Captura da tela aprimorada de busca de atividades executadas.

5.4.5 Apresentação do Resultado para os Participantes

Com o objetivo de permitir que a ferramenta fosse utilizada para treinamento, foi adicionada uma tela com uma tabela de comparação entre os valores inseridos pelo grupo de estimativa e os esforços reais, como exibido na Figura 31, com os valores calculados de diferença relativa e diferença absoluta de acordo com as fórmulas (6) e (7). Esta tela foi adicionada após a realização dos experimentos e apenas pode ser acessada, assim como seu link no menu lateral, se o grupo de participante tiver encerrado as estimativas e a ferramenta estiver configurada para permitir este acesso.

Time: "Time 1" [Banco de dados histórico](#) [Sair do experimento](#)

Projeto

Objetivos do projeto: Portal de transparência para gastos municipais.

Tecnologias empregadas: PHP 5.4, MySQL 5.3, HTML 4, CSS 2, jQuery 1.8.

Tamanho estimado do projeto: 1170 homens-hora.

Requisitos a estimar

Requisito 1	Abrir
Requisito 2	Abrir
Requisito 3	Abrir
Requisito 4	Abrir
Requisito 5	Abrir
Requisito 6	Abrir
Requisito 7	Abrir
Requisito 8	Abrir
Requisito 9	Abrir
Requisito 10	Abrir
Requisito 11	Abrir
Requisito 12	Abrir
Requisito 13	Abrir
Requisito 14	Abrir
Requisito 15	Abrir
Requisito 16	Abrir
Requisito 17	Abrir

Resultados

Tabela de Comparação

Requisito	Estimativa final	Esforço real	Diferença relativa (%)	Diferença absoluta (AR)
1	1.0	2.16	46.3	1.16
2	2.0	2.19	91.32	0.19
3	3.0	1.78	168.54	1.22
4	4.0	1.14	350.88	2.86
5	5.0	0.56	892.86	4.44
6	6.0	0.48	1250.0	5.52
7	7.0	2.38	294.12	4.62
8	8.0	0.3	2666.67	7.7
9	9.0	0.45	2000.0	8.55
10	10.0	0.31	3225.81	9.69
11	11.0	12.23	89.94	1.23
12	12.0	0.18	6666.67	11.82
13	13.0	0.36	3611.11	12.64
14	14.0	1.91	732.98	12.09
15	15.0	1.88	797.87	13.12
16	16.0	4.2	380.95	11.8
17	17.0	0.44	3863.64	16.56

Resultados

[Tabela de Comparação](#)

Figura 31 – Captura da tela com a tabela de comparação do resultado do grupo de estimadores.

$$DR_i = \left| \frac{EE_i}{ER_i} \right| \times 100 \quad (6)$$

$$AR_i = |ER_i - EE_i| \quad (7)$$

Na fórmula (6), DR_i representa a diferença relativa da atividade i , EE_i

representa o esforço estimado da atividade i , e ER_i representa o esforço real da atividade i . Já na fórmula (7), AR_i representa a diferença absoluta da atividade i , EE_i representa o esforço estimado da atividade i , e ER_i representa o esforço real da atividade i .

5.5 FERRAMENTA DE ANÁLISE DE DADOS

Enquanto a ferramenta de coleta de dados apenas acessa os dados de uma etapa do experimento por vez, a ferramenta de análise de dados acessa os dados de todos os bancos e dados de dados coletados, comparando-os com os valores ideais e uns com os outros.

5.5.1 Análise de Melhoria do Grupo de Estimadores

As páginas acessadas na área de análise de melhoria do grupo de estimadores focam no resultado dos mesmos, sendo estes comparados com os valores ideais e entre si, em ambas as etapas e entre elas. A primeira destas páginas é a lista de grupos, ou times, de estimadores, como exibido na Figura 33, onde é possível acessar a comparação entre eles ao clicar em “Tela de comparação” ou a cada um isoladamente, ao clicar no nome do grupo de estimadores.

5.5.1.1 Análise de Melhoria de Acurácia do Grupo de Estimadores Entre Etapas

Nesta tela, primeiramente, são exibidos os nomes dos participantes e as datas em que os experimentos foram conduzidos, exibindo assim mudanças de membros do grupo de estimadores e o intervalo entre as etapas do experimento, como na Figura 32.

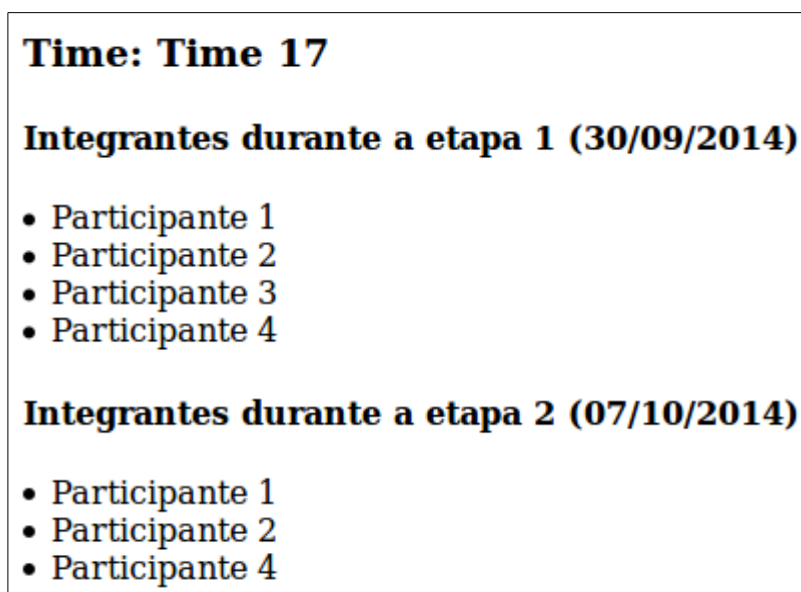


Figura 32 – Captura da tela de análise de melhoria de acurácia do grupo de estimadores entre etapas com os dados de apresentação do grupo de estimadores selecionado.

Na mesma tela, logo abaixo dos dados de apresentação do grupo de estimadores, são exibidos os resultados por etapa do experimento, com os valores apresentados, como na Figura 34, utilizando as fórmulas (6) para diferença relativa, (7) para diferença absoluta (AR), (8) para erro quadrático médio (MSE), (9) para erro relativo balanceado (BRE), (10) para erro relativo balanceado invertido (IBRE), (11) para magnitude do erro relativo a estimativa (MER), e (12) para magnitude do erro relativo (MRE), além de exibir as médias e medianas para cada uma das fórmulas e apresentar o PRED(25), que exibe quantos valores de MRE ficaram abaixo de 0,25 relativo ao número de estimativas realizadas, como na fórmula (13). Como não há consenso entre os autores da área de estimativa para uma fórmula que defina a acurácia das estimativas, os resultados apresentados nesta ferramenta utilizam todas as fórmulas encontradas nas pesquisas da literatura, permitindo assim a comparação direta com qualquer um destes estudos.

Times

Tela de Comparação

Experimento 1

- [Time 1](#)
- [Time 2](#)
- [Time 3](#)
- [Time 4](#)
- [Time 5](#)

Experimento 2

- [Time 6](#)
- [Time 7](#)
- [Time 8](#)
- [Time 9](#)
- [Time 10](#)
- [Time 11](#)
- [Time 12](#)
- [Time 13](#)
- [Time 14](#)

Experimento 3

- [Time 15](#)
- [Time 16](#)
- [Time 17](#)
- [Time 18](#)

Experimento 4

- [Time 19](#)
- [Time 20](#)
- [Time 21](#)
- [Time 22](#)
- [Time 23](#)
- [Time 24](#)

Figura 33 – Captura da tela da lista de grupos de estimadores a selecionar para análise de resultados.

	Estimativa final	Esforço real	Diferença relativa (%)	Diferença absoluta (AR)	Erro quadrático médio (MSE)	Erro relativo balanceado (BRE)
Requisito 1 (42468397)	10.0	2.16	462.96	7.84	61.47	3.63
Requisito 2 (126262945)	7.0	2.19	319.63	4.81	23.14	2.2
Requisito 3 (144640061)	8.0	1.78	449.44	6.22	38.69	3.49
Requisito 4 (194581800)	12.0	1.14	1052.63	10.86	117.94	9.53
Requisito 5 (214999048)	3.0	0.56	535.71	2.44	5.95	4.36
Requisito 6 (266550291)	2.0	0.48	416.67	1.52	2.31	3.17
Requisito 7 (288684360)	4.0	2.38	168.07	1.62	2.62	0.68
Requisito 8 (442586268)	2.0	0.3	666.67	1.7	2.89	5.67
Requisito 9 (449054388)	1.0	0.45	222.22	0.55	0.3	1.22
Requisito 10 (626030107)	1.0	0.31	322.58	0.69	0.48	2.23
Requisito 11 (700897373)	1.0	0.18	555.56	0.82	0.67	4.56
Requisito 12 (737136106)	1.0	0.36	277.78	0.64	0.41	1.78
Requisito 13 (769436331)	2.0	1.91	104.71	0.09	0.01	0.05
Requisito 14 (902919597)	15.0	1.88	797.87	13.12	172.13	6.98
Requisito 15 (955133900)	15.0	4.2	357.14	10.8	116.64	2.57
Requisito 16 (1001054915)	2.0	0.44	454.55	1.56	2.43	3.55

Figura 34 – Captura parcial de tela com a tabela de análise de melhoria de acurácia do grupo de estimadores entre etapas com os resultados de uma etapa isolada.

$$MSE_i = (x_i - \hat{x}_i)^2 \quad (8)$$

$$BRE_i = \frac{|x_i - \hat{x}_i|}{\min(x_i, \hat{x}_i)} \quad (9)$$

$$IBRE_i = \frac{|x_i - \hat{x}_i|}{\max(x_i, \hat{x}_i)} \quad (10)$$

$$MER_i = \frac{|x_i - \hat{x}_i|}{x_i} \quad (11)$$

$$MRE_i = \frac{|x_i - \hat{x}_i|}{\hat{x}_i} \quad (12)$$

$$PRED(25) = \frac{100}{ne} \times \sum_{i \in MRE} \begin{cases} se\ MRE_i \leq 0.25 & 1 \\ senão & 0 \end{cases} \quad (13)$$

Nas fórmulas (8), (9), (10), (11), e (12), i representa o número da estimativa, x_i representa o esforço real, e \hat{x}_i representa o esforço estimado. Já na fórmula (13), ne representa o número de estimativas, i representa o número da estimativa, e MRE_i representa a magnitude do erro relativo para cada i estimativa.

Ainda na mesma tela, abaixo das tabelas de resultados por etapa do experimento, são comparados os resultados das etapas em uma tabela de análise de melhoria de acurácia, como na Figura 35, utilizando as fórmulas (14) para melhora entre as etapas, (15) para melhora relativa ao real, (17) para aumento relativo da diferença absoluta, (18) para aumento relativo do erro quadrático médio, (19) para aumento relativo do erro relativo balanceado, (20) para aumento relativo do erro relativo balanceado invertido, (21) para aumento relativo da magnitude do erro relativo a estimativa, e (22) para aumento relativo da magnitude do erro relativo.

	Etapa 1 Estimativa final	Etapa 2 Estimativa final	Esforço real	Melhora entre as etapas	Melhora relativa ao real	Aumento relativo (%) da diferença absoluta (AR)	Aumento relativo (%) do erro quadrático médio (MSE)
Requisito 1	10.0	4.0	2.16	150.0	277.78	-76.53	-94.49
Requisito 2	7.0	2.0	2.19	250.0	228.31	-96.05	-99.84
Requisito 3	8.0	6.0	1.78	33.33	112.36	-32.15	-53.97
Requisito 4	12.0	5.0	1.14	140.0	614.04	-64.46	-87.37
Requisito 5	3.0	2.0	0.56	50.0	178.57	-40.98	-65.17
Requisito 6	2.0	4.0	0.48	-100.0	-416.67	131.58	436.29
Requisito 7	4.0	4.0	2.38	0.0	0.0	0.0	0.0
Requisito 8	2.0	2.0	0.3	0.0	0.0	0.0	0.0
Requisito 9	1.0	0.5	0.45	100.0	111.11	-90.91	-99.17
Requisito 10	1.0	0.5	0.31	100.0	161.29	-72.46	-92.42
Requisito 11	1.0	1.0	0.18	0.0	0.0	0.0	0.0
Requisito 12	1.0	0.5	0.36	100.0	138.89	-78.13	-95.21
Requisito 13	2.0	0.5	1.91	-300.0	-78.53	1466.67	24444.44
Requisito 14	15.0	4.0	1.88	275.0	585.11	-83.84	-97.39

Figura 35 – Captura parcial de tela com a tabela de análise de melhoria de acurácia do grupo de estimadores entre etapas com a comparação dos resultados de cada etapa.

$$MEAE_i = \left(\frac{\max\left(\frac{\hat{x}_i}{x_i}, \frac{\hat{y}_i}{x_i}\right)}{\min\left(\frac{\hat{x}_i}{x_i}, \frac{\hat{y}_i}{x_i}\right)} - 1 \right) \times \begin{cases} se |x_i - \hat{x}_i| > |x_i - \hat{y}_i| & 100 \\ \text{senão} & -100 \end{cases} \quad (14)$$

$$MRR_i = \left| \frac{\hat{x}_i}{x_i} - \frac{\hat{y}_i}{x_i} \right| \times \begin{cases} se |x_i - \hat{x}_i| > |x_i - \hat{y}_i| & 100 \\ \text{senão} & -100 \end{cases} \quad (15)$$

$$DREE(x, y) = \begin{cases} se x > y & \frac{\min(x, y)}{\max(x, y)} - 1 \\ \text{senão} & \frac{\max(x, y)}{\min(x, y)} - 1 \end{cases} \times 100 \quad (16)$$

$$ARDA_i = DREE(AR_i, \widehat{AR}_i) \quad (17)$$

$$AREQM_i = DREE(MSE_i, \widehat{MSE}_i) \quad (18)$$

$$ARERB_i = DREE(BRE_i, \widehat{BRE}_i) \quad (19)$$

$$ARERBI_i = DREE(IBRE_i, \widehat{IBRE}_i) \quad (20)$$

$$ARMERE_i = DREE(MER_i, \widehat{MER}_i) \quad (21)$$

$$ARMER_i = DREE(MRE_i, \widehat{MRE}_i) \quad (22)$$

Nas fórmulas (14), e (15), i representa o número da estimativa, x_i representa o esforço real, \hat{x}_i representa o esforço estimado na primeira etapa, \hat{y}_i representa o esforço estimado na segunda etapa. Para a fórmula (16), $DREE$ representa a diferença relativa entre etapas, e onde x e y representam os valores na primeira e na segunda etapa, respectivamente. Já as fórmulas (17), (18), (19), (20), (21), e (22), utilizam a fórmula (16), informando os como parâmetros os valores calculados para cada duas etapas, com as fórmulas (7), (8), (9), (10), (11), e (12), respectivamente.

5.5.1.2 Comparação da Melhoria de Acurácia entre os Grupos de Estimadores

Na tela de comparação da melhoria de acurácia entre os grupos de estimadores, são exibidos, resumidamente, os participantes que mudaram entre as etapas, e a contagem das estimativas cuja acurácia melhorou, piorou, ou não mudou, como na Figura 36.

Experimento	Time	Participantes na etapa 1	Participantes da etapa 1 que saíram do time	Participantes que entraram no time na etapa 2	Estimativas comparáveis	Resultado
1	1: Time 1	3	3	0	0	Indiferente
1	2: Time 2	3	1	2	8	Piorou
1	3: Time 3	3	0	1	3	Melhorou
1	4: Time 4	3	0	1	6	Melhorou
1	5: Time 5	2	2	0	0	Indiferente
2	6: Time 6	3	0	0	16	Melhorou
2	7: Time 7	3	0	0	10	Piorou
2	8: Time 8	3	0	0	16	Indiferente
2	9: Time 9	4	0	0	9	Melhorou
2	10: Time 10	3	0	0	10	Melhorou
2	11: Time 11	3	0	0	10	Melhorou

Figura 36 – Captura parcial de tela com a tabela de comparação de melhoria de acurácia do grupo de estimadores entre grupos de estimadores

Para definir se uma estimativa melhorou, piorou, ou se manteve, ao comparar as estimativas entre a primeira etapa e a segunda, foi aplicado o Erro

Quadrático Médio (MSE), a fórmula (8), para cada estimativa em ambas as etapas, e:

- As estimativas cujos valores diminuíram entre a primeira e a segunda etapa foram contadas como “Melhoria”.
- As estimativas cujos valores aumentaram entre a primeira e a segunda etapa foram contadas como “Piora”.
- As estimativas cujos valores não mudaram, com duas casas de precisão, foram contadas como “Não Mudaram”.

5.5.2 Análise de Melhoria por Requisitos

As páginas acessadas na área de análise de melhoria por requisito focam no resultado dos grupos de estimadores, porém separando por requisito, com o propósito de encontrar resultados divergentes entre os requisitos. Na primeira destas páginas são exibidos os links, como exibido na Figura 37, para as comparações realizadas, analisando diferentes aspectos. As páginas acessadas pelos links são descritas neste capítulo.

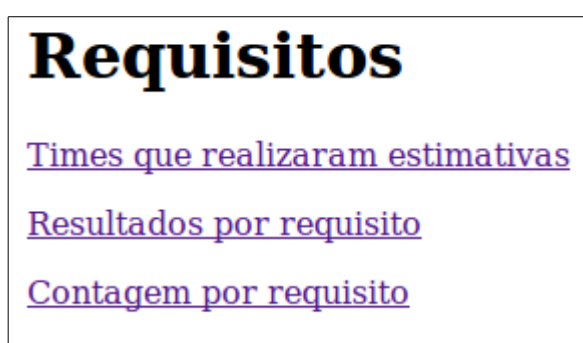


Figura 37 – Captura da tela de links de comparações de resultado por requisito.

5.5.2.1 Lista de Times que Responderam cada Requisito

Nesta tela, são exibidos os nomes dos times, separados por experimento, para cada requisito, exibindo se o time estimou ou não requisito, para cada requisito, como na Figura 38. Desta forma é possível acompanhar o andamento das estimativas.

Requisitos				
Times que realizaram estimativas				
Times que realizaram estimativa para os requisitos em ambas as etapas				
	Experimento 1	Experimento 2	Experimento 3	Experimento 4
Requisito 1	<input checked="" type="checkbox"/> Time 1 <input type="checkbox"/> Time 2 <input type="checkbox"/> Time 3 <input checked="" type="checkbox"/> Time 4 <input type="checkbox"/> Time 5	<input checked="" type="checkbox"/> Time 6 <input checked="" type="checkbox"/> Time 7 <input checked="" type="checkbox"/> Time 8 <input checked="" type="checkbox"/> Time 9 <input checked="" type="checkbox"/> Time 10 <input checked="" type="checkbox"/> Time 11 <input checked="" type="checkbox"/> Time 12 <input checked="" type="checkbox"/> Time 13 <input checked="" type="checkbox"/> Time 14	<input checked="" type="checkbox"/> Time 15 <input checked="" type="checkbox"/> Time 16 <input checked="" type="checkbox"/> Time 17 <input checked="" type="checkbox"/> Time 18	<input type="checkbox"/> Time 19 <input checked="" type="checkbox"/> Time 20 <input checked="" type="checkbox"/> Time 21 <input checked="" type="checkbox"/> Time 22 <input checked="" type="checkbox"/> Time 23 <input checked="" type="checkbox"/> Time 24
Requisito 2	<input checked="" type="checkbox"/> Time 1 <input type="checkbox"/> Time 2 <input type="checkbox"/> Time 3 <input checked="" type="checkbox"/> Time 4 <input checked="" type="checkbox"/> Time 5	<input checked="" type="checkbox"/> Time 6 <input checked="" type="checkbox"/> Time 7 <input checked="" type="checkbox"/> Time 8 <input checked="" type="checkbox"/> Time 9 <input checked="" type="checkbox"/> Time 10 <input checked="" type="checkbox"/> Time 11 <input checked="" type="checkbox"/> Time 12 <input checked="" type="checkbox"/> Time 13 <input checked="" type="checkbox"/> Time 14	<input checked="" type="checkbox"/> Time 15 <input checked="" type="checkbox"/> Time 16 <input checked="" type="checkbox"/> Time 17 <input checked="" type="checkbox"/> Time 18	<input checked="" type="checkbox"/> Time 19 <input checked="" type="checkbox"/> Time 20 <input checked="" type="checkbox"/> Time 21 <input checked="" type="checkbox"/> Time 22 <input checked="" type="checkbox"/> Time 23 <input checked="" type="checkbox"/> Time 24

Figura 38 – Captura da tela de lista de times que responderam cada requisito.

5.5.2.2 Análise de Resultados por Requisito

Nesta tela, primeiramente, são exibidas as médias e medianas dos resultados dos grupos de estimadores por etapa e requisito, como nas Tabelas 1, 2 e 3, de modo a permitir a descoberta de resultados divergentes ao focar no requisito ao invés do grupo de estimadores.

Tabela 1 – Tabela de análise de resultados por requisito para uma etapa, parte 1.

Requisito	Esforço Real	Esforço Estimado Médio	Esforço Estimado Mediano	Diferença Absoluta Média (MAR)	Diferença Absoluta Mediana (MdAR)	Erro Quadrático Médio Médio (MMSE)
1 (42468397)	2.16	9.78	8.0	7.69	5.84	126.42
2 (126262945)	2.19	3.47	3.0	1.8	1.19	6.77
3 (144640061)	1.78	6.79	5.0	5.1	3.22	51.02
4 (194581800)	1.14	9.41	8.0	8.33	6.86	106.87
5 (214999048)	0.56	2.43	2.0	1.91	1.44	7.88
6 (266550291)	0.48	2.84	1.8	2.44	1.32	13.44
7 (288684360)	2.38	7.05	4.0	5.12	1.63	62.04
8 (442586268)	0.3	1.25	1.0	0.99	0.7	1.87
9 (449054388)	0.45	1.16	0.5	0.88	0.3	2.56
10 (626030107)	0.31	1.44	1.0	1.15	0.69	4.74
11 (700897373)	0.18	0.54	0.5	0.42	0.32	0.32
12 (737136106)	0.36	0.29	0.13	0.29	0.26	0.11
13 (769436331)	1.91	1.58	1.0	1.16	0.91	1.78
14 (902919597)	1.88	5.37	4.7	3.96	2.82	28.21
15 (955133900)	4.2	7.41	5.0	4.54	2.7	35.06
16 (1001054915)	0.44	0.59	0.5	0.46	0.35	0.36

Tabela 2 – Tabela de análise de resultados por requisito para uma etapa, parte 2.

Requisito	Erro Quadrático Médio Mediano (MdMSE)	Erro Relativo Balanceado Médio (MBRE)	Erro Relativo Balanceado Mediano (MdBRE)	Erro Relativo Balanceado Invertido Médio (MIBRE)	Erro Relativo Balanceado Invertido Mediano (MdIBRE)
1 (42468397)	34.11	3.57	2.7	0.67	0.73
2 (126262945)	1.42	1.18	0.83	0.39	0.45
3 (144640061)	10.37	2.88	1.81	0.62	0.64
4 (194581800)	47.06	7.34	6.02	0.82	0.86
5 (214999048)	2.07	3.45	2.57	0.63	0.72
6 (266550291)	1.74	5.25	3.17	0.7	0.76
7 (288684360)	2.66	2.25	1.1	0.51	0.52
8 (442586268)	0.49	3.5	2.33	0.64	0.7
9 (449054388)	0.09	2.53	1.22	0.49	0.55
10 (626030107)	0.48	2.53	2.23	0.59	0.69
11 (700897373)	0.1	2.7	1.78	0.62	0.64
12 (737136106)	0.07	4.58	2.19	0.6	0.68
13 (769436331)	0.83	1.46	0.91	0.5	0.48
14 (902919597)	7.95	2.41	1.69	0.61	0.63
15 (955133900)	7.29	1.26	1.1	0.48	0.52
16 (1001054915)	0.12	2.48	1.27	0.59	0.56

Tabela 3 – Tabela de análise de resultados por requisito para uma etapa, parte 3.

Requisito	Magnitude do Erro Relativo a Estimativa Média (MMER)	Magnitude do Erro Relativo a Estimativa Mediana (MdMER)	Magnitude do Erro Relativo Média (MMRE)	Magnitude do Erro Relativo Mediana (MdMRE)	PRED(25)
1 (42468397)	3.56	2.7	0.68	0.73	4.0
2 (126262945)	0.82	0.54	0.75	0.45	28.0
3 (144640061)	2.86	1.81	0.63	0.64	8.0
4 (194581800)	7.3	6.02	0.86	0.86	0.0
5 (214999048)	3.42	2.57	0.67	0.72	8.0
6 (266550291)	5.08	2.75	0.87	0.76	0.0
7 (288684360)	2.15	0.68	0.62	0.7	24.0
8 (442586268)	3.29	2.33	0.84	0.7	8.0
9 (449054388)	1.96	0.67	1.05	0.55	28.0
10 (626030107)	3.72	2.23	0.62	0.69	10.0
11 (700897373)	2.32	1.78	1.0	0.64	7.0
12 (737136106)	0.79	0.72	4.39	1.92	16.0
13 (769436331)	0.61	0.48	1.35	0.91	7.0
14 (902919597)	2.1	1.5	0.92	0.73	7.0
15 (955133900)	1.08	0.64	0.66	0.65	18.0
16 (1001054915)	1.04	0.8	2.04	0.78	18.0

Na mesma tela, logo abaixo das tabelas de médias e medianas dos resultados dos grupos de estimadores por etapa e requisito, são comparados os aumentos relativos dos resultados entre as etapas. Os dados comparados são os mesmos exibidos nas Tabelas 1, 2 e 3.

5.5.2.3 Contagem de Resultados por Requisito

Nesta tela, são exibidos os resultados de forma simplificada, contando o número de resultados positivos, negativos e indiferentes encontrados, ao comparar o Erro Quadrático Médio (MSE) das etapas, fórmula (8), assim como o valor relativo ao total de estimativas desta estimativa, como apresentado na Tabela 4.

Tabela 4 – Tabela de contagem de resultados dos grupos de estimadores por requisito.

Requisito	Resultado	Quantas Melhoraram	Quantas Melhoraram (%)	Quantas Se mantiveram	Quantas Se mantiveram (%)	Quantas Pioraram	Quantas Pioraram (%)
1 (42468397)	Melhorou	13	65,0	1	5,0	6	30,0
2 (126262945)	Melhorou	9	40,91	5	22,73	8	36,36
3 (144640061)	Melhorou	12	57,14	3	14,29	6	28,57
4 (194581800)	Melhorou	14	66,67	3	14,29	4	19,05
5 (214999048)	Melhorou	14	66,67	1	4,76	6	28,57
6 (266550291)	Piorou	7	36,84	3	15,79	9	47,37
7 (288684360)	Melhorou	9	45,0	4	20,0	7	35,0
8 (442586268)	Melhorou	10	50,0	2	10,0	8	40,0
9 (449054388)	Melhorou	9	47,37	6	31,58	4	21,05
10 (626030107)	Indiferente	7	46,67	1	6,67	7	46,67
11 (700897373)	Melhorou	7	58,33	1	8,33	4	33,33
12 (737136106)	Melhorou	5	45,45	4	36,36	2	18,18
13 (769436331)	Indiferente	5	41,67	2	16,67	5	41,67
14 (902919597)	Melhorou	9	75,0	1	8,33	2	16,67
15 (955133900)	Melhorou	4	36,36	4	36,36	3	27,27
16 (1001054915)	Melhorou	9	81,82	1	9,09	1	9,09

5.5.3 Análise de Melhoria por Participante

As páginas acessadas na área de análise de melhoria por participante focam no resultado da estimativa particular de cada estimador participante, com o propósito de encontrar resultados divergentes entre os participantes e aqueles que têm maior poder de convencimento. Na primeira destas páginas são exibidos os links, como exibido na Figura 39. As páginas acessadas pelos links são descritas neste capítulo.

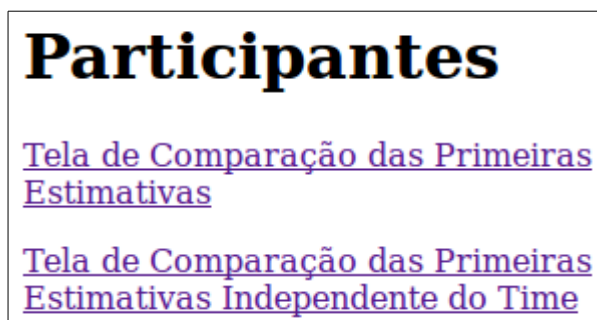


Figura 39 – Captura da tela de links de contagem de resultado por participante.

5.5.3.1 Contagem de Melhoria por Participante Nos Times

Nesta tela, exibida parcialmente na Figura 40, foram contadas as estimativas que melhoraram, pioraram, ou não mudaram, entre a primeira e a segunda etapa, para cada experimento, grupo de estimadores, e primeira estimativa individual de cada estimador participante.

Comparações das Primeiras Estimativas						
Experimento	Time	Participante	Estimativas Comparáveis	Resultado	Quantas Melhoraram	Quantas Melhoraram (%)
1	1: Time 1	Participante 1	0	Indiferente	0	0,0
1	1: Time 1	Participante 2	0	Indiferente	0	0,0
1	1: Time 1	Participante 3	7	Melhorou	5	71,43
1	2: Time 2	Participante 4	0	Indiferente	0	0,0
1	2: Time 2	Participante 5	6	Melhorou	4	66,67
1	2: Time 2	Participante 6	8	Piorou	0	0,0
	2:					

Figura 40 – Captura parcial de tela de análise de melhoria de acurácia por estimador participante individual em cada grupo.

A estimativa individual escolhida foi a primeira dada, já que seguindo o *Planning Poker*, seria a estimativa individual informada com menos influência do grupo.

Na tabela exibida parcialmente na Figura 40, as colunas exibidas, com valores de exemplo, são:

- Experimento: 1.
- Time: 1: Analistas de Software.
- Participante: Ciclano.
- Estimativas Comparáveis: 7.
- Resultado: Melhorou.
- Quantas Melhoraram: 5.

- Quantas Melhoraram (%): 71,43.
- Quantas Se Mantiveram: 0.
- Quantas Se Mantiveram (%): 0,0.
- Quantas Pioraram: 2.
- Quantas Pioraram (%): 28,57.

5.5.3.2 Contagem de Melhoria por Participante Independentes dos Times

Nesta tela, exibida parcialmente na Figura 41, foram contadas as estimativas que melhoraram, pioraram, ou não mudaram, entre a primeira e a segunda etapa, para cada experimento, e primeira estimativa individual de cada estimador participante. Diferente da tela anterior, exibida parcialmente na Figura 40, esta ignora em que grupo de estimadores a estimativa foi realizada, para levar em conta os estimadores participantes que mudaram de grupo de estimadores entre uma etapa e outra.

Tela de Comparação das Primeiras Estimativas Independente do Time						
Experimento	Participante	Estimativas Comparáveis	Resultado	Quantas Melhoraram	Quantas Melhoraram (%)	Quantas Se Mantiveram
1	Participante 1	0	Indiferente	0	0,0	0
1	Participante 2	6	Melhorou	4	66,67	0
1	Participante 3	1	Melhorou	1	100,0	0
1	Participante 4	1	Melhorou	1	100,0	0
1	Participante 5	0	Indiferente	0	0,0	0
1	Participante 6	0	Indiferente	0	0,0	0
1	Participante 7	0	Indiferente	0	0,0	0
1	Participante 8	8	Piorou	0	0,0	3

Figura 41 – Captura parcial de tela de análise de melhoria de acurácia por estimador participante individual independente do grupo de estimadores.

Na tabela exibida parcialmente na Figura 41, as colunas exibidas, com valores de exemplo, são:

- Experimento: 1.
- Participante: Ciclano.
- Estimativas Comparáveis: 7.
- Resultado: Melhorou.
- Quantas Melhoraram: 5.
- Quantas Melhoraram (%): 71,43.
- Quantas Se Mantiveram: 0.
- Quantas Se Mantiveram (%): 0,0.
- Quantas Pioraram: 2.

- Quantas Pioraram (%): 28,57.

5.5.4 Análise de Correlação

As páginas acessadas na área de análise de correlação focam na busca de correlação entre os dados coletados e calculados, exibindo os coeficientes de correlação de Pearson Error: Reference source not found. Na primeira destas páginas são exibidos os links, como exibido na Figura 42. As páginas acessadas pelos links são descritas neste capítulo.

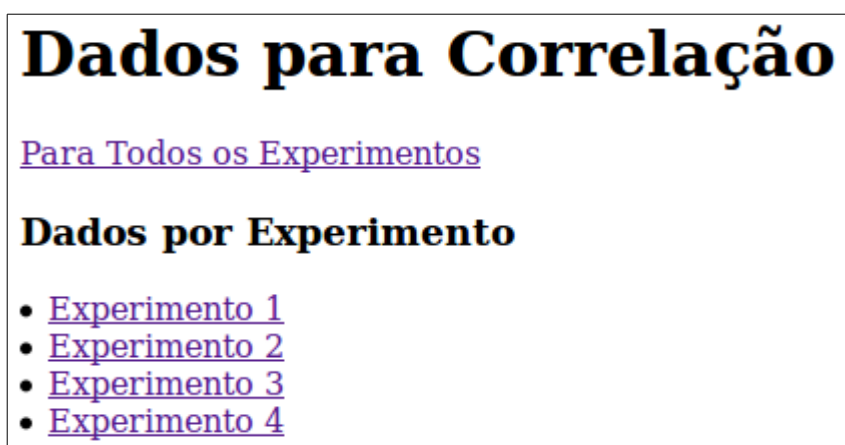


Figura 42 – Captura da tela de links de dados de correlação.

5.5.4.1 Dados de Correlação para Todos Experimentos

São exibidas duas tabelas e um quadro, uma de dados processados, para correlação manual, uma tabela com os resultados de correlação entre as variáveis definidas no capítulo 4.5, e, por fim, o quadro com a legenda das siglas utilizadas na tabela de resultados da busca de correlação.

A primeira tabela exhibe os dados, para cada time:

- Número do Experimento.
- Número do Grupo de Estimadores.

- Nome do Grupo de Estimadores.
- Para Cada Etapa:
 - Integrantes da Equipe.
 - Média das Notas.
 - Média da Diferença Absoluta (MAR).
 - Média do Erro Quadrático Médio (MMSE).
 - Média do Erro Relativo Balanceado (MBRE).
 - Média do Erro Relativo Balanceado Invertido (MIBRE).
 - Média da Magnitude do Erro Relativo a Estimativa (MMER).
 - Média da Magnitude do Erro Relativo (MMRE).
 - Mediana da Diferença Absoluta (MdAR).
 - Mediana do Erro Quadrático Médio (MdMSE).
 - Mediana do Erro Relativo Balanceado (MdBRE).
 - Mediana do Erro Relativo Balanceado Invertido (MdIBRE).
 - Mediana da Magnitude do Erro Relativo a Estimativa (MdMER).
 - Mediana da Magnitude do Erro Relativo (MdMRE).
 - PRED(25).
- Aumento relativo (%) de:
 - Média da Diferença Absoluta (MAR).
 - Média do Erro Quadrático Médio (MMSE).
 - Média do Erro Relativo Balanceado (MBRE).
 - Média do Erro Relativo Balanceado Invertido (MIBRE).
 - Média da Magnitude do Erro Relativo a Estimativa (MMER).
 - Média da Magnitude do Erro Relativo (MMRE).
 - Mediana da Diferença Absoluta (MdAR).
 - Mediana do Erro Quadrático Médio (MdMSE).
 - Mediana do Erro Relativo Balanceado (MdBRE).
 - Mediana do Erro Relativo Balanceado Invertido (MdIBRE).
 - Mediana da Magnitude do Erro Relativo a Estimativa (MdMER).
 - Mediana da Magnitude do Erro Relativo (MdMRE).
 - PRED(25).

- Estimativas que Melhoram (%).
- Estimativas que Não Mudaram (%).
- Estimativas que Pioraram (%).

Já na segunda tabela, apresentada na Figura 43, são exibidos os valores calculados da correlação para cada uma das variáveis consideradas, mais claramente no quadro da Figura 44. Nesta tabela, vários valores devem ser desconsiderados por não terem relevância, como a correlação entre a quantidade relativa de estimativas que melhoraram e as que pioraram.

	MNPMC	ARMDA	AREQM	NEM	NEP	MNP	MMDAE1
ARMDA	0,22						
AREQM	0,23	0,98					
NEM	-0,12	-0,69	-0,6				
NEP	0,16	0,79	0,69	-0,81			
MNP	-0,02	-0,06	0,02	0,46	-0,33		
MMDAE1	-0,32	-0,26	-0,22	0,4	-0,3	0,19	
MMDAE2	0,0	0,71	0,67	-0,54	0,66	-0,12	0,36

Figura 43 – Captura da tabela de resultados de testes de correlação de dados dos experimentos.

Legenda	
Sigla	Descrição
MNPMC	Média das notas dos participantes em matérias de computação
ARMDA	Aumento relativo (%) da média da diferença absoluta (MAR)
AREQM	Aumento relativo (%) do erro quadrático médio (MSE)
NEM	Número de estimativas que melhoram (%)
NEP	Número de estimativas que pioraram (%)
MNP	Média do número de participantes
MMDAE1	Média da média da diferença absoluta (MAR) da etapa 1
MMDAE2	Média da média da diferença absoluta (MAR) da etapa 2

Figura 44 – Captura do quadro de legenda de siglas usadas na exibição de resultados de testes de correlação de dados dos experimentos.

5.5.4.2 Dados de Correlação para Um Experimento Específico

É exibida uma tela bastante similar à descrita no capítulo 5.5.4.1, porém, sem exibir a coluna de número do experimento, e utilizando apenas os dados do experimento específico.

6 EXPERIMENTO

O objetivo deste capítulo é apresentar como foram executados os experimentos com 79 alunos da Universidade Tecnológica Federal do Paraná (UTFPR), dos cursos de Sistemas de Informação e de Engenharia de Computação, cursando as disciplinas de Análise e Projeto de Sistemas ou Engenharia de Software 2.

Foram realizados quatro experimentos similares, cada um em duas etapas, de acordo com a metodologia desenvolvida, apresentada nas Figuras 3, 4 e 5, e com a técnica de realização de estimativa *Planning Poker*, como definido na Figura 1, para a primeira etapa, e adicionando a revisão de atividades executadas, como definido na Figura 2, para segunda etapa. As condições que variaram na aplicação dos experimentos estão descritas no Quadro 14.

Experimento	Iteração 1				Iteração 2			
	1	2	3	4	3	4	3	4
Curso	Engenharia de Computação		Sistemas de Informação		Sistemas de Informação		Engenharia de Computação	
Disciplina cursada pelos alunos participantes	Análise e Projeto de Sistemas		Engenharia de Software 2		Análise e Projeto de Sistemas		Análise e Projeto de Sistemas	
Data da Introdução a <i>Planning Poker</i>	13/08/2014		18/08/2014		30/09/2014		29/10/2014	
Etapas	1	2	1	2	1	2	1	2
Data de realização da etapa	20/08/14	27/08/14	25/08/14	25/08/14	30/09/14	07/10/14	29/10/14	26/11/14
Número de participantes na etapa	14	12	28	28	15	13	20	22

Quadro 14 – Variações das conduções dos experimentos.

Na primeira etapa, os participantes tiveram uma introdução à técnica *Planning Poker*, para assim serem capazes de realizar o experimento, conforme as Figuras 3, 4 e 5 e, em grupos de três a quatro pessoas, realizaram estimativas de

esforço em homens/hora da implementação de 17 requisitos reais de um ambiente de desenvolvimento de software, oriundos de empresa comercial, sendo um deles posteriormente desconsiderado por ter esforço realizado em desacordo com sua complexidade. Na segunda etapa, os participantes foram apresentados ao mesmo conjunto de requisitos, porém desta vez tendo acesso a outras especificações de requisitos semelhantes, com seus esforços reais apresentados. Esforços reais que também foram estimados utilizando *Planning Poker*, e realizados em ambiente de desenvolvimento da mesma empresa.

Analisando os experimentos 1 e 2 realizados, respectivamente, com os cursos de Engenharia de Computação e Sistemas de Informação, verificou-se a necessidade de adicionar dois novos passos na segunda iteração:

- Pesquisadores guiam alguns participantes em uma demonstração do processo de *Planning Poker*: incluído para melhorar a compreensão dos participantes através de um exemplo prático, sem citar valores ou interpretações dos participantes, como na Figura 3. Durante os experimentos 1 e 2, alguns alunos interrompiam o processo de estimativa de *Planning Poker* chamando os pesquisadores pois, embora tivessem facilidade em utilizar a ferramenta de software, não haviam entendido bem os passos de execução da técnica.
- Cada participante preenche o formulário de avaliação de usabilidade e de satisfação: incluído para obter o ponto de vista dos participantes sobre o uso da ferramenta de coleta. Permitindo assim verificar problemas de usabilidade e aprimorar a ferramenta de coleta em próximas versões, como na Figura 5.

Para colher os dados de cada estimativa, foi utilizada uma ferramenta web desenvolvida para isto, esta descrita no capítulo 5.4, onde as justificativas e estimativas individuais geradas na discussão realizada durante o *Planning Poker* foram armazenadas, como por exemplo, complexidade, dificuldade da linguagem de programação, profundidade insuficiente dos requisitos, entre outros. Além destes, foram colhidos dados para verificação de correlação, como experiência acadêmica, experiência profissional, confiança em suas habilidades de programação, design de software, análise de requisitos, entre outros.

6.1 DADOS COLETADOS NO EXPERIMENTO

Ao comparar as estimativas finais com os valores de esforço reais, foi possível apontar os times que obtiveram melhora na estimativa. Porém, vários outros fatores foram considerados como a mudança dos participantes no time entre as etapas e a realização das etapas em dias diferentes. Além do fato de que vários times não estimaram todos os requisitos, fazendo com que a análise dos resultados tivesse que descartar aqueles que não foram realizados em ambas as etapas. O resultado da comparação inicial pode ser vista na Tabela 5.

Tabela 5 – Comparação inicial dos times em seus experimentos.

Experimen- to	Time	Participantes na etapa 1	Participantes da etapa 1 que saíram do time	Participantes que entraram no time na eta- pa 2	Estimativas comparáveis	Resultado
1	1	3	3	0	0	Não considerado
1	2	3	1	2	8	Piorou
1	3	3	0	1	3	Não considerado
1	4	3	0	1	6	Melhorou
1	5	2	2	0	0	Não considerado
2	6	3	0	0	16	Melhorou
2	7	3	0	0	10	Piorou
2	8	3	0	0	16	Indiferente
2	9	4	0	0	9	Melhorou
2	10	3	0	0	10	Melhorou
2	11	3	0	0	10	Melhorou
2	12	3	0	0	10	Piorou
2	13	3	0	0	16	Melhorou
2	14	3	0	0	16	Melhorou
3	15	4	2	1	14	Indiferente
3	16	3	0	0	15	Melhorou
3	17	4	1	0	15	Melhorou
3	18	4	0	0	16	Melhorou
4	19	4	0	0	16	Melhorou
4	20	3	0	0	16	Piorou
4	21	3	0	1	5	Melhorou
4	22	3	1	1	8	Melhorou
4	23	3	0	0	16	Melhorou
4	24	4	0	1	16	Melhorou

Tiveram de ser desconsiderados os times 1 e 5, pela falta de seus participantes na segunda etapa do experimento, e o time 3 pela falta de estimativas cadastradas.

Após a definição dos times que seriam considerados, foi aplicada a fórmula de Erro Quadrático Médio (MSE), fórmula (8), para cada estimativa em ambas as etapas, e:

- Para os valores que diminuiram, com duas casas de precisão, entre a

primeira e a segunda etapa, foram contados como “Melhoria”.

- Para os valores que aumentaram, com duas casas de precisão, entre a primeira e a segunda etapa, foram contados como “Piora”.

- Para os valores que não mudaram, com duas casas de precisão, foram contados como “Não Mudaram”.

O resultado geral dos experimentos por time é exibida com detalhes na Tabela 6. A fórmula (8), de Erro Quadrático Médio (MSE), foi escolhida para esta tarefa, por desconsiderar alterações mínimas, como um centésimo de hora, diferente das outras fórmulas específicas para avaliação de estimativa.

Tabela 6 – Comparação da melhoria de acurácia dos times em seus experimentos.

Experi- mento	Time	Total de Estimati- vas	Quantas melhora- ram	Quantas melhora- ram (%)	Quantas não mu- daram	Quantas não muda- ram (%)	Quantas pioraram	Quantas pioraram (%)
1	2	8	0	0,0	0	0,0	8	100,0
1	4	6	4	66,67	0	0,0	2	33,33
2	6	16	9	56,25	2	12,5	5	31,25
2	7	10	2	20,0	0	0,0	8	80,0
2	8	16	7	43,75	2	12,5	7	43,75
2	9	9	9	100,0	0	0,0	0	0,0
2	10	10	3	30,0	6	60,0	1	10,0
2	11	10	8	80,0	1	10,0	1	10,0
2	12	10	3	30,0	1	10,0	6	60,0
2	13	16	6	37,5	5	31,25	5	31,25
2	14	16	9	56,25	3	18,75	4	25,0
3	15	14	7	50,0	0	0,0	7	50,0
3	16	15	10	66,67	2	13,33	3	20,0
3	17	15	9	60,0	2	13,33	4	26,67
3	18	16	10	62,5	4	25,0	2	12,5
4	19	16	13	81,25	3	18,75	0	0,0
4	20	16	4	25,0	1	6,25	11	68,75
4	21	5	3	60,0	2	40,0	0	0,0
4	22	8	3	37,5	3	37,5	2	25,0
4	23	16	11	68,75	2	12,5	3	18,75
4	24	16	11	68,75	3	18,75	2	12,5

6.2 ANÁLISE DOS RESULTADOS

Ao comparar as estimativas finais com os valores de esforço reais, foi possível apontar os grupos que obtiveram melhora na estimativa. Porém, dos 24 grupos que participaram da pesquisa, foram desconsiderados os grupos 1, 2, 5, 15, 17 e 22 pela falta de seus participantes na segunda etapa do experimento, os grupos 2, 3, 4, 15, 21, 22 e 24 por incluírem componentes no grupo na segunda etapa do experimento e os grupos 3, 4, 5, 21 por terem analisado menos da metade do total dos 16 requisitos analisados. Ou seja, para este estudo, foram analisados apenas os dados dos 14 grupos de alunos que se mantiveram com os mesmos componentes em ambas as etapas e analisaram mais da metade dos requisitos propostos para os experimentos. O resultado da comparação pode ser visto na Tabela 7. Após a definição dos grupos que seriam considerados, foi aplicada a fórmula da Média dos valores de Diferença Absoluta Média (MAR) da primeira etapa e da segunda etapa. Sendo MAR a média das Diferenças Absolutas (AR) das estimativas, fórmula (7).

Tabela 7 – Comparação da melhoria de acurácia dos grupos entre as etapas.

Iteração	Experimento	Grupo	MDA1 (Etapa 1)	MDA2 (Etapa 2)	Melhoria das estimativas entre as etapas 1 e 2(MRMDA)
1	2	6	3,46	3,4	1,73
1	2	7	1,51	3,67	-143,05
1	2	8	3,26	2,12	34,97
1	2	9	11,06	3,28	70,34
1	2	10	2,98	1,83	38,59
1	2	11	6,05	2,03	66,45
1	2	12	7,93	6,83	13,87
1	2	13	0,82	1,0	-21,95
1	2	14	2,67	1,43	46,44
2	3	16	1,15	0,48	58,26
2	3	18	1,95	0,99	49,23
2	4	19	2,35	0,66	71,91
2	4	20	2,68	3,77	-40,67
2	4	23	2,06	0,72	65,05

Na Tabela 7, quando há melhoria nas estimativas, ou seja, a diferença entre o valor real o estimado diminui entre etapas, MRMDA é positivo, sendo negativo quando houve aumento na diferença. Os valores para MRMDA resultam da equação (23).

$$MRMDA = -DREE(MDA_1, MDA_2) \quad (23)$$

Na fórmula (23), *MRMDA* representa a melhoria das estimativas entre as etapas, e *MDA₁* e *MDA₂* representa as médias das Diferenças Absolutas para a primeira e segunda etapas, respectivamente. A função *DREE* foi apresentada como a fórmula (16).

Considerando apenas estes mesmos grupos, ao verificar os dados apresentados na Tabela 6, percebe-se que, dos 14 grupos analisados:

- 8 grupos tiveram uma melhora da acurácia das estimativas maior que 50%.
- Em 3 grupos, a soma das estimativas com melhora e das estimativas que não mudaram foi maior que 50%.
- 3 grupos tiveram uma piora da acurácia das estimativas maior que 50%.

Ao verificar as possíveis correlações levantadas no capítulo 4.5, foram encontrados os resultados apresentados na Figura 45. E embora se esperasse uma correlação entre a média das notas dos participantes, em disciplinas da área de computação, e o aumento relativo (%) da média da diferença absoluta (MAR), esta não foi encontrada, já que seu valor de correlação (negativo) foi de -0,21. Na Figura 46, é apresentado o gráfico de dispersão relativo às variáveis citadas na correlação analisada.

Tabela de Resultado de Testes de Correlação dos Experimentos

	MNPMC	ARMDA	AREQM	NEM	NEP	MNP	MMDAE1
ARMDA	0,21						
AREQM	0,26	0,98					
NEM	-0,02	-0,75	-0,63				
NEP	0,12	0,85	0,77	-0,81			
MNP	0,02	-0,38	-0,32	0,61	-0,54		
MMDAE1	-0,31	-0,3	-0,3	0,4	-0,18	0,29	
MMDAE2	-0,08	0,42	0,35	-0,42	0,62	-0,2	0,58

Legenda

Sigla	Descrição
MNPMC	Média das notas dos participantes em matérias de computação
ARMDA	Aumento relativo (%) da média da diferença absoluta (MAR)
AREQM	Aumento relativo (%) do erro quadrático médio (MSE)
NEM	Número de estimativas que melhoram (%)
NEP	Número de estimativas que pioraram (%)
MNP	Média do número de participantes
MMDAE1	Média da média da diferença absoluta (MAR) da etapa 1
MMDAE2	Média da média da diferença absoluta (MAR) da etapa 2

Figura 45 – Captura da tabela de resultados de testes de correlação de dados dos experimentos para os grupos selecionados.

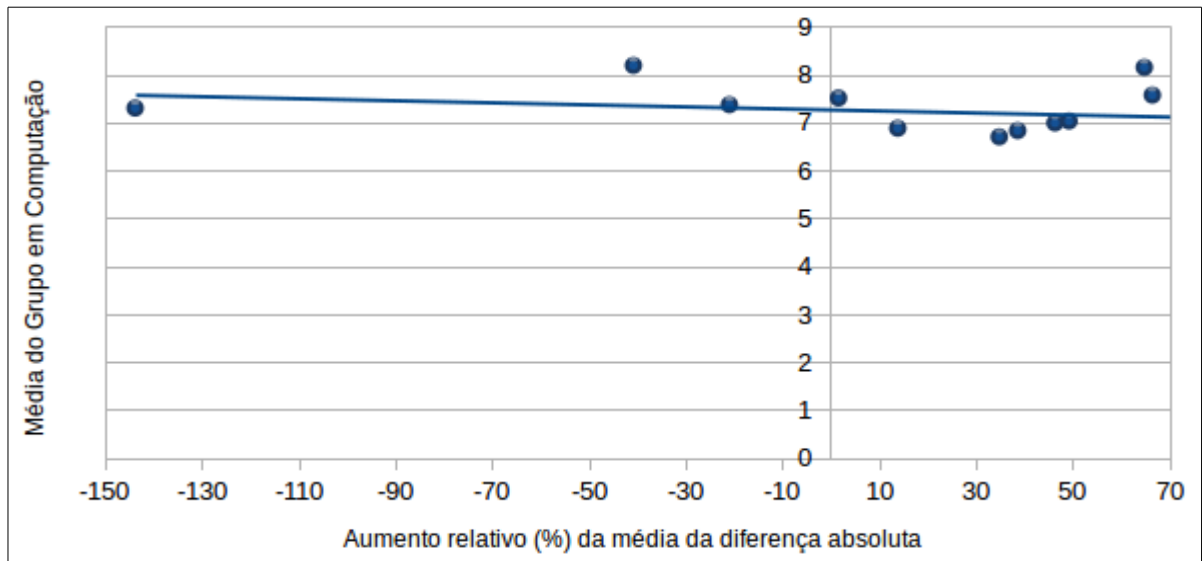


Figura 46 – Gráfico de dispersão entre a média das notas de cada grupo e o aumento relativo da média da diferença absoluta.

7 AVALIAÇÃO DE USABILIDADE DA FERRAMENTA

A usabilidade é um importante indicador de qualidade de software, pois onde a usabilidade é adequada, o usuário tem maior facilidade em apreender a realizar as operações, com menos erros e dificuldades (Yan e Guo 2010).

7.1 QUESTIONÁRIO

Para avaliar a ferramenta, um questionário foi desenvolvido, revisado e aplicado ao final dos terceiro e quarto experimentos. O questionário tem respostas fechadas baseadas, principalmente, na escala de Likert (Allen e Seaman 2007), sendo apresentado, no formato em que foi aplicado, no Apêndice C. Para as questões que utilizaram a escala de Likert, as opções “Concordo plenamente”, “Concordo parcialmente”, “Indiferente”, “Discordo parcialmente”, e “Discordo completamente” foram oferecidas.

7.2 RESULTADOS DO QUESTIONÁRIO DE AVALIAÇÃO DA FERRAMENTA DE COLETA DE DADOS

As respostas das questões de usabilidade e satisfatibilidade foram lidas e compiladas, respectivamente, nas Tabelas 8 e 9.

Tabela 8 – Respostas do questionário de usabilidade.

Questão	Concordo plenamente (%)	Concordo parcialmente (%)	Indiferente (%)	Discordo parcialmente (%)	Discordo completamente (%)
1	86,7	10,0	3,3	0,0	0,0
2	70,0	26,7	0,0	3,3	0,0
3	83,3	16,7	0,0	0,0	0,0
4	40,0	43,3	3,3	3,3	0,0
5	70,0	30,0	0,0	0,0	0,0
6	83,3	16,7	0,0	0,0	0,0
7	43,3	33,3	6,7	13,3	3,3
8	40,0	50,0	3,3	6,7	0,0
9	43,3	33,3	13,3	10,0	0,0
10	40,0	43,3	6,7	10,0	0,0
11	93,3	3,3	3,3	0,0	0,0
12	93,3	3,3	3,3	0,0	0,0
13	13,3	36,7	43,3	3,3	3,3
14	13,3	33,3	40,0	10,0	3,3
15	93,3	6,7	0,0	0,0	0,0
16	40,0	43,3	3,3	3,3	10,0
17	13,3	36,7	13,3	36,7	0,0
18	70,0	16,7	10,0	3,3	0,0
19	86,7	10,0	3,3	0,0	0,0
20	46,7	40,0	13,3	0,0	0,0
21	66,7	23,3	10,0	0,0	0,0
22	73,3	13,3	13,3	0,0	0,0

Tabela 9 – Respostas do questionário de satisfatibilidade.

Questão	Concordo plenamente (%)	Concordo parcialmente (%)	Indiferente (%)	Discordo parcialmente (%)	Discordo completamente (%)
1	66,7	26,7	6,7	0,0	0,0
2	90,0	6,7	3,3	0,0	0,0
3	20,0	50,0	30,0	0,0	0,0
4	23,3	43,3	33,3	0,0	0,0

7.3 ANÁLISE DA USABILIDADE

As respostas fornecidas pelos participantes do último experimento, apresentados no capítulo 7.2, foram analisadas e para os casos cujo resultado não foi tão positivo quanto esperado, foram comentadas neste capítulo.

O aspecto que sofreu maior rejeição foi a atratividade do leiaute que, por ser simplista, não foi considerado visualmente agradável e com pouca variação na forma de apresentação dos dados. Este leiaute foi uma decisão de design da ferramenta desde sua modelagem que, por evitar usar cores, apenas para mensagens de erro, componentes visuais personalizados, ou que não foram fornecidos nativamente pelo navegador, foi dada mais importância à compatibilidade entre navegadores e o contraste e clareza nos textos apresentados. Porém, como a ferramenta foi desenvolvida com estrutura e estilo separados, é de fácil alteração de código-fonte, necessitando apenas da definição de um design mais atrativo.

Houve grande indiferença dos participantes nas questões 13 e 14 do questionário de usabilidade. Estas questões abordam a relação entre o plano de fundo utilizado e, respectivamente, o tema e o propósito da ferramenta. E pode ter se mostrado desta forma pela escolha visual simplista do plano de fundo branco em todas as telas. Embora esta escolha não tenha causado problemas, há espaço para melhorias neste aspecto.

Os erros de grafia e gramática apontados por alguns dos participantes se devem aos textos de requisitos, muitas vezes menos claros que poderiam ser ou então contendo erros de grafia ou gramática. Porém como erros assim são comuns no dia a dia de desenvolvimento, inclusive por misturar português e inglês, foram apenas corrigidos, dos requisitos vindos do banco de dados legado, os erros que comprometiam a compreensão.

Os resultados do questionário de satisfatibilidade deixou claro o interesse em realizar a estimativa de software e na técnica de estimativa *Planning Poker*, porém apenas dois terços dos participantes que responderam o questionário utilizariam a ferramenta pela sua praticidade ou eficácia.

8 CONCLUSÃO

Este trabalho apresenta um método proposto, denominado Revisão de Atividades Executadas utilizando *Planning Poker*. Tal método foi testado com alunos de graduação, dos Cursos de Sistemas de Informação e Engenharia de Computação da UTFPR, por meio de sua implementação em uma ferramenta web. Ferramenta esta que, criada para seguir o *Planning Poker*, reforça a correta aplicação da técnica, permite o armazenamento de dados gerados na discussão, e facilita a revisão de estimativas antigas em base histórica.

Dos 24 grupos que participaram dos experimentos, foi possível manter 14 grupos com os mesmos componentes nas etapas 1 e 2 dos experimentos, devido aos alunos que faltaram ou na etapa 1 ou na etapa 2 de cada experimento. Esse fato representou uma grande dificuldade para efetuar os experimentos e também deve ser considerado como uma limitação do experimento, devendo ser levado em conta por pesquisadores que realizam experimentos com alunos de graduação em um meio acadêmico.

Dessa forma, entre as limitações do estudo, podem-se destacar a falta de experiência profissional dos participantes, o que exigiu a apresentação do método em detalhes antes da execução dos experimentos, e a irregularidade da configuração de alguns grupos, que comprometeu resultados que foram desconsiderados no capítulo de resultados e discussão.

De acordo com os experimentos, pode-se observar que o método de Revisão de Atividades Executadas utilizando *Planning Poker*, melhorou a estimativa de esforço da maioria dos grupos analisados, podendo ser um importante método para aprimorar o processo de desenvolvimento de software e para treinamento de alunos e profissionais na área de estimativa de software.

A melhoria da acurácia foi encontrada com o método porque o mesmo permite que os estimadores utilizem estimativas de atividades já executadas que julguem similares e, além disso, que foquem o seu julgamento apenas nas diferenças entre as atividades e projetos encontrados e os que desejam estimar, partindo de valores de referência, no lugar de depender estritamente de suas capacidades de imaginar ou de lembrar de eventos passados de implementação de requisitos.

Praticamente não houve correlação entre a média das notas de cada grupo, relativas a disciplinas da área de Computação que os alunos cursaram, e o aumento relativo (%) da média da diferença absoluta (MAR), com correlação (negativa) $r = -0,21$. Ou seja, o grau de aproveitamento em disciplinas anteriormente cursadas na área de Computação influenciou negativamente na melhora das estimativas entre as etapas.

Com base nos pontos fracos observados, na existência de poucos trabalhos na literatura neste contexto e devido aos métodos mais comuns na indústria de software para realização de estimativas serem os baseados em julgamento de especialista (Jørgensen 2013b), foi desenvolvido um novo método e uma ferramenta que pode ser utilizado também no ensino de estimativa de software, sendo aplicados em contextos reais de ensino, melhorando a qualidade na formação dos alunos nesta área.

Como trabalhos futuros poderiam ser realizados trabalhos com outros métodos e técnicas além do *Planning Poker* e incluindo o armazenamento das argumentações das discussões coletadas através da voz, ao invés do uso do teclado, o que poderia agilizar o processo. Poderiam ser buscadas correlações entre outros valores coletados ou calculados, em níveis diferentes de granularidade. Também poderiam ser realizados experimentos com estimadores geograficamente distribuídos, profissionais com diferentes níveis de experiência em estimativa, desenvolvimento e gerência de equipes, além de aprimorar sua interface, tornando-o mais produtivo e agradável.

REFERÊNCIAS

- Ahmed, M.A., and Muzaffar, Z. (2009). Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic based framework. *Information and Software Technology* 51, pp. 640–654.
- Ahmed, M.A., Ahmad, I., and AlGhamdi, J.S. (2013). Probabilistic size proxy for software effort prediction: A framework. *Information and Software Technology* 55, pp. 241–251.
- Albrecht, A.J., and Gaffney, J.E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *Software Engineering, IEEE Transactions on SE-9*, pp. 639–648.
- Allen, I.E., and Seaman, C.A. (2007). Likert Scales and Data Analyses. *Quality Progress* 40, pp. 64.
- Aroonvatanaporn, P., Koolmanojwong, S., and Boehm, B. (2012). COTIPMO: A CONstructive Team Improvement Process MOdel. In *Software and System Process (ICSSP), 2012 International Conference on*, pp. 28–37.
- Basten, D., and Sunyaev, A. (2011). Guidelines for software development effort estimation. *Computer* 44, pp. 88–90.
- Benediktsson, O., Dalcher, D., Reed, K., and Woodman, M. (2003). COCOMO-Based Effort Estimation for Iterative and Incremental Software Development. *Software Quality Journal* 11, pp. 471–484.
- Berlin, S., Raz, T., Glezer, C., and Zviran, M. (2009). Comparison of estimation methods of cost and duration in IT projects. *Information and Software Technology* 51, pp. 738–748.
- Bibi, S., Stamelos, I., and Angelis, L. (2008). Combining probabilistic models for explanatory productivity estimation. *Information and Software Technology* 50, pp. 656–669.
- Boehm, B., and Valerdi, R. (2011). Impact of Software Resource Estimation Research on Practice: A Preliminary Report on Achievements, Synergies, and Challenges. In *Proceedings of the 33rd International Conference on Software Engineering*, (New York, NY, USA: ACM), pp. 1057–1065.
- Børte, K., Ludvigsen, S.R., and Mørch, A.I. (2012). The role of social interaction in software effort estimation: Unpacking the “magic step” between reasoning and decision-making. *Information and Software Technology* 54, pp. 985–996.
- Buglione, L., and Ebert, C. (2011). Estimation Tools and Techniques. *IEEE Softw.* 28, pp. 91–94.
- Castro-Herrera, C., and Cleland-Huang, J. (2010). Utilizing Recommender Systems to Support Software Requirements Elicitation. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*, (New York, NY, USA: ACM), pp. 6–10.
- Choudhari, J., and Suman, U. (2012). Phase Wise Effort Estimation for Software Maintenance: An Extended SMEEM Model. In *Proceedings of the CUBE*

- International Information Technology Conference, (New York, NY, USA: ACM), pp. 397–402.
- Cohn, M. (2005). Agile estimating and planning (Prentice Hall Professional Technical Reference).
- Dagnino, A. (2013). Estimating software-intensive projects in the absence of historical data. In Software Engineering (ICSE), 2013 35th International Conference on, pp. 941–950.
- Dejaeger, K., Verbeke, W., Martens, D., and Baesens, B. (2012). Data mining techniques for software effort estimation: A comparative study. *IEEE Transactions on Software Engineering* 38, pp. 375–397.
- Desharnais, J.-M., Buglione, L., and Kocatürk, B. (2011). Using the COSMIC Method to Estimate Agile User Stories. In Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, (New York, NY, USA: ACM), pp. 68–73.
- Feitosa, A.C., and Campos, G.M.M. (2010). AprendES: um jogo educacional para auxiliar o processo de ensino-aprendizagem da Engenharia de Software. (João Pessoa, Brasil), pp. 1–4.
- Grenning, J. (2002). Planning poker or how to avoid analysis paralysis while release planning. Disponível em <http://www.renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf>. Acesso em: 27 ago. 2015.
- Gruschke, T.M., and Jørgensen, M. (2008). The Role of Outcome Feedback in Improving the Uncertainty Assessment of Software Development Effort Estimates. *ACM Trans. Softw. Eng. Methodol.* 17, pp. 1–35.
- Guarino, S.L., Pfautz, J.D., Cox, Z., and Roth, E. (2009). Modeling human reasoning about meta-information. *International Journal of Approximate Reasoning* 50, pp. 437–449.
- Hale, M., Jorgenson, N., and Gamble, R. (2011). Predicting individual performance in student project teams. In Software Engineering Education and Training (CSEE T), 2011 24th IEEE-CS Conference on, pp. 11–20.
- Hassouna, A., and Tahvildari, L. (2010). An effort prediction framework for software defect correction. *Information and Software Technology* 52, pp. 197–209.
- Haugen, N.C. (2006). An empirical study of using planning poker for user story estimation. In Agile Conference, 2006, pp. 9–34.
- Hearty, P., Fenton, N., Marquez, D., and Neil, M. (2009). Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. *IEEE Trans. Softw. Eng.* 35, pp. 124–137.
- Heikkilä, V., Jadallah, A., Rautiainen, K., and Ruhe, G. (2010). Rigorous Support for Flexible Planning of Product Releases - A Stakeholder-Centric Approach and Its Initial Evaluation. In System Sciences (HICSS), 2010 43rd Hawaii International Conference on, pp. 1–10.
- Hericko, M., and Živkovič, A. (2008). The size and effort estimates in iterative development. *Information and Software Technology* 50, pp. 772–781.
- Huang, S.-J., Chiu, N.-H., and Liu, Y.-J. (2008). A comparative evaluation on the

- accuracies of software effort estimates from clustered data. *Information and Software Technology* 50, pp. 879–888.
- Jani, H.M. (2010). Applying Case-Based Reasoning to software requirements specifications quality analysis system. In *Software Engineering and Data Mining (SEDM), 2010 2nd International Conference on*, pp. 140–144.
- Jørgensen, M. (2009). How to avoid selecting bids based on overoptimistic cost estimates. *IEEE Software* 26, pp. 79–84.
- Jørgensen, M. (2010). Identification of more risks can lead to increased over-optimism of and over-confidence in software development effort estimates. *Information and Software Technology* 52, pp. 506–516.
- Jørgensen, M. (2011). Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation. *Information and Software Technology* 53, pp. 1382–1390.
- Jørgensen, M. (2013a). The influence of selection bias on effort overruns in software development projects. *Information and Software Technology* 55, pp. 1640–1650.
- Jørgensen, M. (2013b). Relative estimation of software development effort: It matters with what and how you compare. *IEEE Software* 30, pp. 74–79.
- Jørgensen, M., and Grimstad, S. (2008). Avoiding Irrelevant and Misleading Information When Estimating Development Effort. *IEEE Softw.* 25, pp. 78–83.
- Jørgensen, M., and Grimstad, S. (2011). The impact of irrelevant and misleading information on software development effort estimates: A randomized controlled field experiment. *IEEE Transactions on Software Engineering* 37, pp. 695–707.
- Jørgensen, M., and Grimstad, S. (2012). Software development estimation biases: The role of interdependence. *IEEE Transactions on Software Engineering* 38, pp. 677–693.
- Jørgensen, M., and Gruschke, T.M. (2009). The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments. *IEEE Trans. Softw. Eng.* 35, pp. 368–383.
- Jørgensen, M., and Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *Software Engineering, IEEE Transactions on* 33, pp. 33–53.
- Jørgensen, M., Boehm, B., and Rifkin, S. (2009). Software Development Effort Estimation: Formal Models or Expert Judgment? *IEEE Softw.* 26, pp. 14–19.
- Keung, J.W., Kitchenham, B.A., and Jeffery, D.R. (2008). Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation. *IEEE Trans. Softw. Eng.* 34, pp. 471–484.
- Kitchenham, B., and Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering (Keele University and Durham University Joint Report).
- Kitchenham, B.A., Mendes, E., and Travassos, G.H. (2007). Cross versus Within-Company Cost Estimation Studies: A Systematic Review. *Software Engineering, IEEE Transactions on* 33, pp. 316–329.
- Kocaguneli, E., Menzies, T., and Keung, J.W. (2012a). On the value of ensemble

- effort estimation. *IEEE Transactions on Software Engineering* 38, pp. 1403–1416.
- Kocaguneli, E., Menzies, T., Bener, A.B., and Keung, J.W. (2012b). Exploiting the essential assumptions of analogy-based effort estimation. *IEEE Transactions on Software Engineering* 38, pp. 425–438.
- Kocaguneli, E., Menzies, T., Keung, J., Cok, D., and Madachy, R. (2013). Active Learning and effort estimation: Finding the essential content of software effort estimation data. *IEEE Transactions on Software Engineering* 39, pp. 1040–1053.
- Kompella, L. (2013). Advancement of Decision-Making in Agile Projects by Applying Logistic Regression on Estimates. In *Global Software Engineering Workshops (ICGSEW), 2013 IEEE 8th International Conference on*, pp. 11–17.
- Lagerström, R., Würtemberg, L., Holm, H., and Luczak, O. (2012). Identifying factors affecting software development cost and productivity. *Software Quality Journal* 20, pp. 395–417.
- Lavazza, L., Morasca, S., and Robiolo, G. (2013). Towards a simplified definition of Function Points. *Information and Software Technology* 55, 1796–1809.
- Mahnič, V. (2012). A Capstone Course on Agile Software Development Using Scrum. *Education, IEEE Transactions on* 55, pp. 99–106.
- Mahnič, V., and Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software* 85, pp. 2086–2095.
- McNely, B.J., Gestwicki, P., Burke, A., and Gelms, B. (2012). Articulating Everyday Actions: An Activity Theoretical Approach to Scrum. In *Proceedings of the 30th ACM International Conference on Design of Communication, (New York, NY, USA: ACM)*, pp. 95–104.
- Mendes, E., and Mosley, N. (2008). Bayesian Network Models for Web Effort Prediction: A Comparative Study. *IEEE Trans. Softw. Eng.* 34, pp. 723–737.
- Menzies, T., Butcher, A., Cok, D., Marcus, A., Layman, L., Shull, F., Turhan, B., and Zimmermann, T. (2013). Local versus global lessons for defect prediction and effort estimation. *IEEE Transactions on Software Engineering* 39, pp. 822–834.
- Minku, L.L., and Yao, X. (2013). Software Effort Estimation As a Multiobjective Learning Problem. *ACM Trans. Softw. Eng. Methodol.* 22, pp. 1–32.
- Miranda, E., Bourque, P., and Abran, A. (2009). Sizing user stories using paired comparisons. *Information and Software Technology* 51, pp. 1327–1337.
- Mittas, N., and Angelis, L. (2013). Ranking and clustering software cost estimation models through a multiple comparisons algorithm. *IEEE Transactions on Software Engineering* 39, pp. 537–551.
- Mittas, N., Athanasiades, M., and Angelis, L. (2008). Improving analogy-based software cost estimation by a resampling method. *Information and Software Technology* 50, pp. 221–230.
- Moe, N.B., Aurum, A., and Dybå, T. (2012). Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology* 54, pp. 853–865.
- Moløkken-Østfold, K., and Jørgensen, M. (2004). Group Processes in Software Effort Estimation. *Empirical Softw. Engg.* 9, pp. 315–334.

- Moløkken-Østfold, K., Haugen, N.C., and Benestad, H.C. (2008). Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software* 81, pp. 2106–2117.
- Muzaffar, Z., and Ahmed, M.A. (2010). Software development effort prediction: A study on the factors impacting the accuracy of fuzzy logic systems. *Information and Software Technology* 52, pp. 92–109.
- Nan, N., and Harter, D.E. (2009). Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort. *IEEE Trans. Softw. Eng.* 35, pp. 624–637.
- Nguyen, V., Boehm, B., and Danphitsanuphan, P. (2011). A controlled experiment in assessing and estimating software maintenance tasks. *Information and Software Technology* 53, pp. 682–691.
- Nguyen-Cong, D., and Tran-Cao, D. (2013). A review of effort estimation studies in agile, iterative and incremental software development. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, 2013 IEEE RIVF International Conference on, pp. 27–30.
- Ochodek, M., Nawrocki, J., and Kwarciak, K. (2011). Simplifying effort estimation based on Use Case Points. *Information and Software Technology* 53, pp. 200–213.
- Oliveira, A.L.I., Braga, P.L., Lima, R.M.F., and Cornélio, M.L. (2010). GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Information and Software Technology* 52, pp. 1155–1166.
- Peixoto, C.E.L., Audy, J.L.N., and Prikladnicki, R. (2010). The Importance of the Use of an Estimation Process. In *Proceedings of the 2010 ICSE Workshop on Software Development Governance*, (New York, NY, USA: ACM), pp. 13–17.
- Pendharkar, P.C., Rodger, J.A., and Subramanian, G.H. (2008). An empirical study of the Cobb–Douglas production function properties of software development effort. *Information and Software Technology* 50, pp. 1181–1188.
- Raith, F., Richter, I., Lindermeier, R., and Klinker, G. (2013). Identification of Inaccurate Effort Estimates in Agile Software Development. In *20th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 67–72.
- Robiolo, G., and Orosco, R. (2007). An Alternative Method Employing Uses Cases for Early Effort Estimation. In *Software Engineering Workshop, 2007. SEW 2007. 31st IEEE*, pp. 89–98.
- R. Santos, Lima, T., Campos, B., and Werner, C. (2013). *Experiência na Avaliação de Efetividade de um Jogo para Ensino de Conceitos de Requisitos em Sistemas Ubíquos*. (Campinas, Brasil), pp. 935–939.
- Seo, Y.-S., Bae, D.-H., and Jeffery, R. (2013). AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning. *Information and Software Technology* 55, pp. 1710–1725.
- Shihab, E., Kamei, Y., Adams, B., and Hassan, A.E. (2013). Is lines of code a good measure of effort in effort-aware models? *Information and Software Technology* 55, pp. 1981–1993.

- Silva, J.C., Sousa, S.P.A., Kulesza, R., and Brito, A.V. (2012). Uma avaliação do emprego do jogo Modelando como apoio ao ensino de Engenharia de Requisitos. (Curitiba, Brasil), pp. 1–11.
- Sommerville, I. (2005). Integrated requirements engineering: a tutorial. *Software, IEEE* 22, pp. 16–23.
- Suelmann, H. (2013). A Comment on C. Symons' "Exploring Software Project Effort versus Duration Trade-offs." *Software, IEEE PP*, pp. 1.
- Symons, C. (2012). Exploring Software Project Effort versus Duration Trade-offs. *IEEE Softw.* 29, pp. 67–74.
- Symons, C.R. (1988). Function point analysis: difficulties and improvements. *Software Engineering, IEEE Transactions on* 14, pp. 2–11.
- Tamrakar, R., and Jørgensen, M. (2012). Does the use of Fibonacci numbers in planning poker affect effort estimates? In *Evaluation Assessment in Software Engineering (EASE 2012), 16th International Conference on*, pp. 228–232.
- Tan, H.B.K., Zhao, Y., and Zhang, H. (2009). Conceptual Data Model-based Software Size Estimation for Information Systems. *ACM Trans. Softw. Eng. Methodol.* 19, pp. 1–37.
- Thiry, M., Zoucas, A., and Gonçalves, R.Q. (2010). Promovendo a Aprendizagem de Engenharia de Requisitos de Software Através de um Jogo Educativo. (João Pessoa, Brasil), pp. 1–10.
- Vargas, D., Moro, T., Dambrósio, G., Cassal, M., Bernardi, G., and Cordenonsi, A.Z. (2010). Desenvolvimento de um Jogo de Empresa baseado em Agentes de Software e Instituições Eletrônicas para simulação de Elicitação de Requisitos de Software. (Belo Horizonte, Brasil), pp. 847–856.
- Wen, J., Li, S., Lin, Z., Hu, Y., and Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology* 54, pp. 41–59.
- Wilkie, F.G., McChesney, I.R., Morrow, P., Tuxworth, C., and Lester, N.G. (2011). The value of software sizing. *Information and Software Technology* 53, pp. 1236–1249.
- Williams, L., Brown, G., Meltzer, A., and Nagappan, N. (2011). Scrum + engineering practices: Experiences of three microsoft teams. In *International Symposium on Empirical Software Engineering and Measurement*, (Banff, AB), pp. 463–471.
- Xia, W., Capretz, L.F., Ho, D., and Ahmed, F. (2008). A new calibration for Function Point complexity weights. *Information and Software Technology* 50, pp. 670–683.
- Yan, P., and Guo, J. (2010). The research of Web usability design. In *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, pp. 480–483.
- Yavari, Y., Afsharchi, M., and Karami, M. (2011). Software complexity level determination using software effort estimation use case points metrics. In *Software Engineering (MySEC), 2011 5th Malaysian Conference in*, pp. 257–262.
- Zhou, Y., Yang, Y., Xu, B., Leung, H., and Zhou, X. (2014). Source code size estimation approaches for object-oriented systems from UML class diagrams: A comparative study. *Information and Software Technology* 56, pp. 220–237.

- Živkovič, A., Gencel, C., and Abran, A. (2011). Guest editorial: Advances in functional size measurement and effort estimation – Extended best papers. *Information and Software Technology* 53, pp. 873.
- Zuppiroli, S., Ciancarini, P., and Gabbrielli, M. (2012). A Role-Playing Game for a Software Engineering Lab: Developing a Product Line. In *Software Engineering Education and Training (CSEE T)*, 2012 IEEE 25th Conference on, pp. 13–22.
- (1998). *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Std 830-1998, pp. 1–40.

APÊNDICE A – Revisões Sistemáticas da Literatura

Com o objetivo de dar base a discussões e descrever o cenário de pesquisa atual, foram realizadas revisões da literatura.

A primeira revisão para compreensão do estado da arte foi realizada no formato de revisão sistemática da literatura examinando o estado da arte da área de pesquisa de Estimativa de Esforço de Software e reportando as conclusões cabíveis.

Já a segunda revisão do estado da arte focou no contexto de *Planning Poker*.

A.1 REVISÃO SISTEMÁTICA DA LITERATURA EM ESTIMATIVA DE ESFORÇO DE SOFTWARE

A pesquisa realizada foi baseada no processo de revisão sistemática da literatura descrito por Kitchenham e Charters (2007). Este processo é dividido em três partes:

- Planejamento da Revisão, no qual o protocolo metodológico de pesquisa é definido.
- Condução da Revisão, no qual o protocolo que foi definido é executado.
- Relatório da Revisão, no qual são relatadas as etapas de revisão a comunidade, apresentadas neste artigo.

A.1.1 Protocolo de Pesquisa de Artigos da Literatura

O protocolo descreve as questões de pesquisa, estratégias de pesquisa, seleção de estudos para inclusão, análise dos estudos selecionados, e síntese dos dados (Kitchenham e Charters 2007). Este capítulo do artigo descreve este protocolo.

A.1.1.1 Questões de Pesquisa de Artigos da Literatura

Ao realizar qualquer revisão sistemática da literatura, é necessária a definição de questões de pesquisa para dirigir a metodologia de pesquisa como um todo (Kitchenham e Charters 2007). As questões de pesquisa para este trabalho são apresentadas nas subseções seguintes.

A.1.1.1.1 Pesquisadores Influentes

Como em qualquer outra área de pesquisa, há pesquisadores que trabalham mais em uma área de pesquisa, influenciando no desenvolvimento de conhecimentos desta área. E para descobrir os pesquisadores mais influentes, foi definida a seguinte questão:

QP1. Quem são os principais pesquisadores da área de estimativas de esforço de desenvolvimento de software?

A.1.1.1.2 Grupos de Pesquisa Influentes

Além dos pesquisadores mais influentes, há certos grupos de pesquisa que aparecem frequentemente, o que sugere o interesse dentre autores em explorarem direções de pesquisa similares. E para descobrir os grupos de pesquisa mais influentes entre os artigos selecionados manualmente, foi definida a seguinte questão:

QP2. Quais são os principais grupos de pesquisa?

A.1.1.1.3 Revistas Influentes

Há revistas em que a pesquisa em estimativa de software aparece mais frequentemente e, portanto denotam interesse na área. E para descobrir as revistas

mais influentes, foi definida a seguinte questão:

QP3. Que revistas costumam publicar artigos na área?

A.1.1.1.4 Principais Abordagens para as Estimativas de Esforço de Software

Há hoje diversas abordagens para se tratar o problema. Porém, com um grande volume de abordagens, se torna necessário compilar as abordagens mais estudadas para se definir quais utilizar. E para descobrir as abordagens principais de estimativa de esforço, foi definida a seguinte questão:

QP4. Quais são as principais abordagens para estimativa de esforço de software?

A.1.1.1.5 Melhoras Significativas Encontradas em Pesquisas

Muitas pesquisas comparativas foram realizadas, porém para encontrar abordagens que demonstrem os melhores resultados, se torna necessário comparar os resultados de pesquisas, obtendo assim as abordagens mais promissoras. E para descobrir estas abordagens de estimativa de esforço mais promissoras, foi definida a seguinte questão:

QP5. Que estudos mostraram melhoras significativas na predição de tempo de execução das atividades de software? E para quais abordagens?

A.1.1.1.6 Tipos de Atividades de Software Cujas Acurácia da Estimativa de Esforço é Menor

Podem-se dividir atividades de software por sua natureza e/ou etapa de realização. Como por exemplo, atividades de desenvolvimento de software, de correção de defeitos, de empacotamento de software, e de testes de software.

Porém, como cada tipo de atividade utiliza conjuntos de habilidades diferentes e obedece a processos de execução diferentes, o índice de acurácia entre elas deve diferir.

Focar em tipos de atividades de software cuja acurácia seja menor pode ser uma forma menos custosa de se aumentar significativamente a acurácia total do projeto. E para descobrir os tipos de atividade de software de menor acurácia de estimativa de esforço, foi definida a seguinte questão:

QP6. Quais os tipos de atividades de software que divergem mais entre o tempo estimado e o tempo realizado?

A.1.1.2 Estratégia de Pesquisa de Artigos da Literatura

Por conta do grande volume de artigos e de não utilizar palavras-chave como parâmetros de pesquisa, foram excluídos os artigos de congresso e patentes. A busca por artigos foi realizada, inicialmente, filtrando-se os artigos por ano de publicação de 2008 a 2013, e por revista, nas bases de dados selecionadas. Após este filtro inicial, os artigos encontrados foram filtrados manualmente por título, resumo e por conteúdo do artigo.

A.1.1.2.1 Seleção de Bases de Dados

Algumas bases de dados de artigos podem influenciar o conjunto de artigos encontrados por beneficiar um formato, um método de estudo ou uma subárea de pesquisa. E, para diminuir esta possível influência, quatro bases de dados foram selecionadas:

- ACM Digital Library.
- IEEE Xplore.
- Scirus.
- Scopus.

A.1.1.2.2 Seleção de Revistas

Com o objetivo de pesquisar artigos relevantes, de revistas relevantes, uma lista de revistas foi definida.

A lista de revistas foi definida com base no resultado de busca de revistas realizado pela ferramenta de busca disponibilizada pelo Journal Citation Reports (JCR).

O JCR é um recurso que avalia os periódicos, por número de citações em uma ou mais áreas de pesquisa e sua relevância para a comunidade científica, atribuindo valores de Fator de Impacto para permitir comparação entre os periódicos.

Apenas as revistas que pertenciam à categoria de Engenharia de Software e que tivessem fator de impacto maior que 1,5 foram escolhidas, e estão apresentadas na Tabela 10.

Tabela 10 – Revistas selecionadas.

Revista	Fator de Impacto (JCR)
ACM Transactions on Graphics	3,361
ACM Transactions on Knowledge Discovery from Data	1,676
ACM Transactions on Mathematical Software	1,934
ACM Transactions on Software Engineering and Methodology	1,548
Communications of the ACM	2,511
Computer	1,675
Computer Graphics Forum	1,638
IEEE Internet Computing	2,039
IEEE Micro	2,386
IEEE Software	1,616
IEEE Transactions on Computational Intelligence and AI in Games	1,694
IEEE Transactions on Multimedia	1,754
IEEE Transactions on Reliability	2,293
IEEE Transactions on Services Computing	2,46
IEEE Transactions on Software Engineering	2,588
IEEE Transactions on Visualization and Computer Graphics	1,898
Image and Vision Computing	1,959
Information and Software Technology	1,522
International Journal of Web and Grid Services	1,615
Journal of Mathematical Imaging and Vision	1,767
Journal of the ACM	2,37
Mathematical Programming	2,09
SIAM Journal on Imaging Sciences	2,966

A.1.1.3 Realização da Pesquisa de Artigos da Literatura

Com o protocolo definido, a pesquisa pode ser realizada. Este capítulo do artigo descreve a realização da pesquisa.

A.1.1.3.1 Seleção de Artigos para Inclusão

A Tabela 11 descreve o número de artigos de cada estágio do processo de seleção.

Como exceção do filtro de busca de ano de publicação, realizado no estágio três, foram adicionados os artigos “*A Systematic Review of Software Development Cost Estimation Studies*” (Jørgensen e Shepperd 2007) e “*Cross versus Within-Company Cost Estimation Studies: A Systematic Review*” (Kitchenham et al. 2007), por suas contribuições na área de pesquisa.

Tabela 11 – Número de artigos avaliados a cada estágio do processo de seleção.

Estágio	Artigos	Adicionados	Total
Estágio 1: filtro por título	24,517	0	24,517
Estágio 2: filtro por resumo	91	0	91
Estágio 3: filtro por conteúdo do artigo	60	2	62
Final: artigos selecionados	51	0	51

A.1.1.3.2 Verificação da Seleção

Durante o primeiro estágio de seleção, pela quantidade de artigos a filtrar manualmente, foram realizadas buscas manuais por dois dos autores deste artigo. As diferenças entre os resultados das buscas foram discutidas. Os resultados apresentados neste artigo foram analisados e acordados entre os autores. Apenas após a conclusão do primeiro estágio de seleção foram removidas as ocorrências repetidas nos resultados da busca.

A.1.1.3.3 Estratégia de Extração de Dados

Para responder as questões QP1, QP2 e QP3, os dados de artigos foram utilizados sem a necessidade de analisá-los individualmente.

Para responder a questão QP4, os artigos foram lidos, e as abordagens relacionadas. Porém, diversas abordagens encontradas foram consideradas similares em sua natureza. Para simplificar a relação entre as abordagens, essas foram agrupadas em categorias, estas definidas na pesquisa realizada por

Jørgensen e Shepperd (2007), o que permitiu a comparação com os resultados encontrados por eles.

Para responder a questão QP5, os artigos foram lidos e, para aqueles com resultados positivos, os resultados encontrados foram agrupados por abordagem.

Para responder a questão QP6, os artigos foram lidos e, para aqueles que classificaram as atividades de software por tipo (natureza e/ou etapa de realização), os índices de acurácia encontrados foram agrupados por tipo.

Os dados dos artigos foram extraídos, classificando-os e apresentando-os como descrito:

- QP1. Os artigos foram agrupados, contando o número de artigos de cada autor, sendo ou não o autor principal. Assim, descobrimos os autores que tem maior presença nesta área de pesquisa.
- QP2. Os autores foram relacionados entre si, e o número de relações foi contado. Assim descobrimos que autores publicam juntos com frequência.
- QP3. Os artigos foram agrupados, contando o número de artigos de cada revista. Assim descobrimos que revistas publicam mais na área de pesquisa.
- QP4. Os artigos foram agrupados por abordagens de estimativa de software que incluem em seus estudos.
- QP5. Os artigos que apresentaram resultados em seus experimentos, e que se mostraram significativamente positivos na aplicação de abordagens de estimativa de software foram agrupados por abordagem.
- QP6. Os artigos que categorizam atividades de software por natureza e/ou etapa de execução foram agrupados e classificados por abordagem e índices de acurácia encontrados.

A.1.1.3.4 Desvios do Protocolo

Nenhum desvio do protocolo foi necessário para conclusão da pesquisa.

A.1.1.3.5 Artigos Seleccionados

Os dados dos artigos seleccionados são apresentados no Quadro 15.

Revista Abreviada	Autores	Ano	Título	Citação
IEEE SW	Suelmann, H.	2013	A Comment on C. Symons' "Exploring Software Project Effort versus Duration Trade-offs"	(Suelmann 2013)
IST	Huang, S.-J.; Chiu, N.-H.; Liu, Y.-J.	2008	A Comparative Evaluation On The Accuracies Of Software Effort Estimates From Clustered Data	(Huang et al. 2008)
IST	Nguyen, V.; Boehm, B.; Danphitsanuphan, P.	2011	A Controlled Experiment In Assessing And Estimating Software Maintenance Tasks	(Nguyen et al. 2011)
IST	Xia, W.; Capretz, L.; Ho, D.; Ahmed, F.	2008	A New Calibration For Function Point Complexity Weights	(Xia et al. 2008)
IEEE TSE	Jørgensen, M.; Shepperd, M.	2007	A Systematic Review of Software Development Cost Estimation Studies	(Jørgensen e Shepperd 2007)
IEEE TSE	Kocaguneli, E.; Menzies, T.; Keung, J.; Cok, D.; Madachy, R.	2013	Active Learning and Effort Estimation: Finding the Essential Content of Software Effort Estimation Data	(Kocaguneli et al. 2013)
IST	Hassouna, A.; Tahvildari, L.	2010	An Effort Prediction Framework For Software Defect Correction	(Hassouna e Tahvildari 2010)
IST	Pendharkar, P.; Rodger, J.; Subramanian, G.	2008	An Empirical Study Of The Cobb-douglas Production Function Properties Of Software Development Effort	(Pendharkar et al. 2008)
IEEE TSE	Keung, J.; Kitchenham, B.; Jeffery, D.	2008	Analogy-X: Providing Statistical Inference to Analogy-Based Software Cost Estimation	(Keung et al. 2008)
IST	Seo, Y.-S.; Bae, D.-H.; Jeffery, R.	2013	AREION: Software Effort Estimation Based On Multiple Regressions With Adaptive Recursive Data Partitioning	(Seo et al. 2013)
IEEE SW	Jørgensen, M.; Grimstad, S.	2008	Avoiding Irrelevant and Misleading Information When Estimating Development Effort	(Jørgensen e Grimstad 2008)
IEEE TSE	Mendes, E.; Mosley, N.	2008	Bayesian Network Models for Web Effort Prediction: A Comparative Study	(Mendes e Mosley 2008)
IST	Bibi, S.; Stamelos, I.; Angelis, L.	2008	Combining Probabilistic Models For Explanatory Productivity Estimation	(Bibi et al. 2008)

Quadro 15 – Lista de artigos selecionados.

Revista Abreviada	Autores	Ano	Título	Citação
IST	Berlin, S.; Raz, T.; Glezer, C.; Zviran, M.	2009	Comparison Of Estimation Methods Of Cost And Duration In It Projects	(Berlin et al. 2009)
ACM TOSEM	Tan, H.; Zhao, Y.; Zhang, H.	2009	Conceptual Data Model-based Software Size Estimation For Information Systems	(Tan et al. 2009)
IST	Jørgensen, M.	2011	Contrasting Ideal And Realistic Conditions As A Means To Improve Judgment-based Software Development Effort Estimation	(Jørgensen 2011)
IEEE TSE	Kitchenham, B.; Mendes, E.; Travassos, G.	2007	Cross versus Within-Company Cost Estimation Studies: A Systematic Review	(Kitchenham et al. 2007)
IEEE TSE	Dejaeger, K.; Verbeke, W.; Martens, D.; Baesens, B.	2012	Data Mining Techniques for Software Effort Estimation: A Comparative Study	(Dejaeger et al. 2012)
IEEE SW	Buglione, L.; Ebert, C.	2011	Estimation Tools and Techniques	(Buglione e Ebert 2011)
IEEE TSE	Kocaguneli, E.; Menzies, T.; Bener, A.; Keung, J.	2012	Exploiting the Essential Assumptions of Analogy-Based Effort Estimation	(Kocaguneli et al. 2012b)
IEEE SW	Symons, C.	2012	Exploring Software Project Effort versus Duration Trade-offs	(Symons 2012)
IST	Oliveira, A.; Braga, P.; Lima, R.; Cornélio, M.	2010	Ga-based Method For Feature Selection And Parameters Optimization For Machine Learning Regression Applied To Software Effort Estimation	(Oliveira et al. 2010)
IST	Živkovič, A.; Gencel, C.; Abran, A.	2011	Guest Editorial: Advances In Functional Size Measurement And Effort Estimation – Extended Best Papers	(Živkovič et al. 2011)
Comp.	Basten, D.; Sunyaev, A.	2011	Guidelines for Software Development Effort Estimation	(Basten e Sunyaev 2011)
IST	Ahmed, M.; Muzaffar, Z.	2009	Handling Imprecision And Uncertainty In Software Development Effort Prediction: A Type-2 Fuzzy Logic Based Framework	(Ahmed e Muzaffar 2009)

Quadro 15 – Lista de artigos selecionados (continuação).

Revista Abreviada	Autores	Ano	Título	Citação
IEEE SW	Jørgensen, M.	2009	How to Avoid Selecting Bids Based on Overoptimistic Cost Estimates	(Jørgensen 2009)
IST	Jørgensen, M.	2010	Identification Of More Risks Can Lead To Increased Over-optimism Of And Over-confidence In Software Development Effort Estimates	(Jørgensen 2010)
IEEE TSE	Nan, N.; Harter, D.	2009	Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort	(Nan e Harter 2009)
IST	Mittas, N.; Athanasiades, M.; Angelis, L.	2008	Improving Analogy-based Software Cost Estimation By A Resampling Method	(Mittas et al. 2008)
IST	Shihab, E.; Kamei, Y.; Adams, B.; Hassan, A.	2013	Is Lines Of Code A Good Measure Of Effort In Effort-aware Models?	(Shihab et al. 2013)
IEEE TSE	Menzies, T.; Butcher, A.; Cok, D.; Marcus, A.; Layman, L.; Shull, F.; Turhan, B.; Zimmermann, T.	2013	Local versus Global Lessons for Defect Prediction and Effort Estimation	(Menzies et al. 2013)
IEEE TSE	Kocaguneli, E.; Menzies, T.; Keung, J.	2012	On the Value of Ensemble Effort Estimation	(Kocaguneli et al. 2012a)
IEEE TSE	Hearty, P.; Fenton, N.; Marquez, D.; Neil, M.	2009	Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model	(Hearty et al. 2009)
IST	Ahmed, M.; Ahmad, I.; Alghamdi, J.	2013	Probabilistic Size Proxy For Software Effort Prediction: A Framework	(Ahmed et al. 2013)
IEEE TSE	Mittas, N.; Angelis, L.	2013	Ranking and Clustering Software Cost Estimation Models through a Multiple Comparisons Algorithm	(Mittas e Angelis 2013)
IEEE SW	Jørgensen, M.	2013	Relative Estimation of Software Development Effort: It Matters with What and How You Compare	(Jørgensen 2013b)
IST	Ochodek, M.; Nawrocki, J.; Kwarciak, K.	2011	Simplifying Effort Estimation Based On Use Case Points	(Ochodek et al. 2011)
IEEE SW	Jørgensen, M.; Boehm, B.; Rifkin, S.	2009	Software Development Effort Estimation: Formal Models or Expert Judgment?	(Jørgensen et al. 2009)

Quadro 15 – Lista de artigos selecionados (continuação).

Revista Abreviada	Autores	Ano	Título	Citação
IST	Muzaffar, Z.; Ahmed, M.	2010	Software Development Effort Prediction: A Study On The Factors Impacting The Accuracy Of Fuzzy Logic Systems	(Muzaffar e Ahmed 2010)
IEEE TSE	Jørgensen, M.; Grimstad, S.	2012	Software Development Estimation Biases: The Role of Interdependence	(Jørgensen e Grimstad 2012)
ACM TOSEM	Minku, L.; Yao, X.	2013	Software Effort Estimation As A Multiobjective Learning Problem	(Minku e Yao 2013)
IST	Zhou, Y.; Yang, Y.; Xu, B.; Leung, H.; Zhou, X.	2013	Source Code Size Estimation Approaches For Object-oriented Systems From UML Class Diagrams: A Comparative Study	(Zhou et al. 2013)
IST	Wen, J.; Li, S.; Lin, Z.; Hu, Y.; Huang, C.	2012	Systematic Literature Review Of Machine Learning Based Software Development Effort Estimation Models	(Wen et al. 2012)
IEEE TSE	Jørgensen, M.; Grimstad, S.	2011	The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experiment	(Jørgensen e Grimstad 2011)
IEEE TSE	Jørgensen, M.; Gruschke, T.	2009	The Impact of Lessons-Learned Sessions on Effort Estimation and Uncertainty Assessments	(Jørgensen e Gruschke 2009)
IST	Jørgensen, M.	2013	The Influence Of Selection Bias On Effort Overruns In Software Development Projects	(Jørgensen 2013a)
ACM TOSEM	Gruschke, T.; Jørgensen, M.	2008	The Role Of Outcome Feedback In Improving The Uncertainty Assessment Of Software Development Effort Estimates	(Gruschke e Jørgensen 2008)
IST	Borte, K.; Ludvigsen, S.; Morch, A.	2012	The Role Of Social Interaction In Software Effort Estimation: Unpacking The "magic Step" Between Reasoning And Decision-making	(Borte et al. 2012)
IST	Hericko, M.; Živkovič, A.	2008	The Size And Effort Estimates In Iterative Development	(Hericko e Živkovič 2008)
IST	Wilkie, F.; McChesney, I.; Morrow, P.; Tuxworth, C.; Lester, N.	2011	The Value Of Software Sizing	(Wilkie et al. 2011)

Quadro 15 – Lista de artigos selecionados (continuação).

Revista Abreviada	Autores	Ano	Título	Citação
IST	Lavazza, L.; Morasca, S.; Robiolo, G.	2013	Towards A Simplified Definition Of Function Points	(Lavazza et al. 2013)

Quadro 15 – Lista de artigos selecionados (continuação).

A.1.2 Resultados e Discussão da Revisão Sistemática da Literatura em Estimativa de Esforço de Software

As questões de pesquisa, que direcionam a pesquisa, são respondidas e discutidas neste capítulo.

A.1.2.1 QP1. Quem São os Principais Pesquisadores da Área de Estimativas de Esforço de Desenvolvimento de Software?

Foram identificados 111 autores dentre os 51 artigos selecionados. A relação de frequência de publicação, exceto os autores que só tiveram uma publicação, está descrita na Tabela 12. E, ao analisar os resultados, pode-se afirmar que:

- Magne Jørgensen foi autor de 23,53% dos artigos selecionados, tornando este o autor com maior presença dentre os artigos selecionados da área de pesquisa em estimativa de software.
- Considerando apenas participações como autor principal, Ekrem Kocaguneli com 5,88% dos artigos selecionados, ocupa a posição de segundo autor mais frequente.
- Considerando também participações como autor não principal, Jacky Wai Keung e Tim Menzies dividem a segunda posição de frequência de publicações com 7,84% dos artigos selecionados.

Tabela 12 – Número de artigos por autor entre os selecionados.

Autor	Artigos como Au- tor	Artigos como Au- tor (%)	Artigos como Au- tor Principal	Artigos como Autor Principal (%)
Ahmed, M.	3	5,88	2	3,92
Angelis, L.	3	5,88	0	0
Boehm, B.	2	3,92	0	0
Cok, D.	2	3,92	0	0
Grimstad, S.	3	5,88	0	0
Gruschke, T.	2	3,92	1	1,96
Jørgensen, M.	12	23,53	11	21,57
Keung, J.	4	7,84	1	1,96
Kitchenham, B.	2	3,92	1	1,96
Kocaguneli, E.	3	5,88	3	5,88
Mendes, E.	2	3,92	1	1,96
Menzies, T.	4	7,84	1	1,96
Mittas, N.	2	3,92	2	3,92
Muzaffar, Z.	2	3,92	1	1,96
Živkovič, A.	2	3,92	1	1,96
Ahmed, M.	3	5,88	2	3,92
Angelis, L.	3	5,88	0	0
Boehm, B.	2	3,92	0	0
Cok, D.	2	3,92	0	0
Grimstad, S.	3	5,88	0	0
Gruschke, T.	2	3,92	1	1,96
Jørgensen, M.	12	23,53	11	21,57
Keung, J.	4	7,84	1	1,96

A.1.2.2 QP2. Quais São os Principais Grupos de Pesquisa?

Ao verificar as relações entre autores, Tim Menzies se destaca por ter participado apoiando estudos de diversos autores, como exibido na Figura 47.

Enquanto que Magne Jørgensen, destaque na questão anterior, mostrou-se trabalhar com um grupo bastante pequeno de pesquisadores, como exibido na Figura 48.

Para as figuras Figura 47 e Figura 48, o número exibido próximo às linhas de ligação é o número de relações encontradas entre os autores correspondentes, e os círculos são as ligações entre todos os ligados a estes. Ao verificar as relações entre

autores, Tim Menzies se destaca por ter participado apoiando estudos de diversos autores, como exibido na Figura 47.

Enquanto que Magne Jørgensen, destaque na questão anterior, mostrou-se trabalhar com um grupo bastante pequeno de pesquisadores, como exibido na Figura 48, assim como os outros autores, como exibido nas Figuras 49, 50 e 51.

Para as Figuras 47, 48, 49, 50 e 51, o número exibido próximo às linhas de ligação é o número de relações encontradas entre os autores correspondentes, e os círculos são as ligações entre todos os autores ligados a estes.

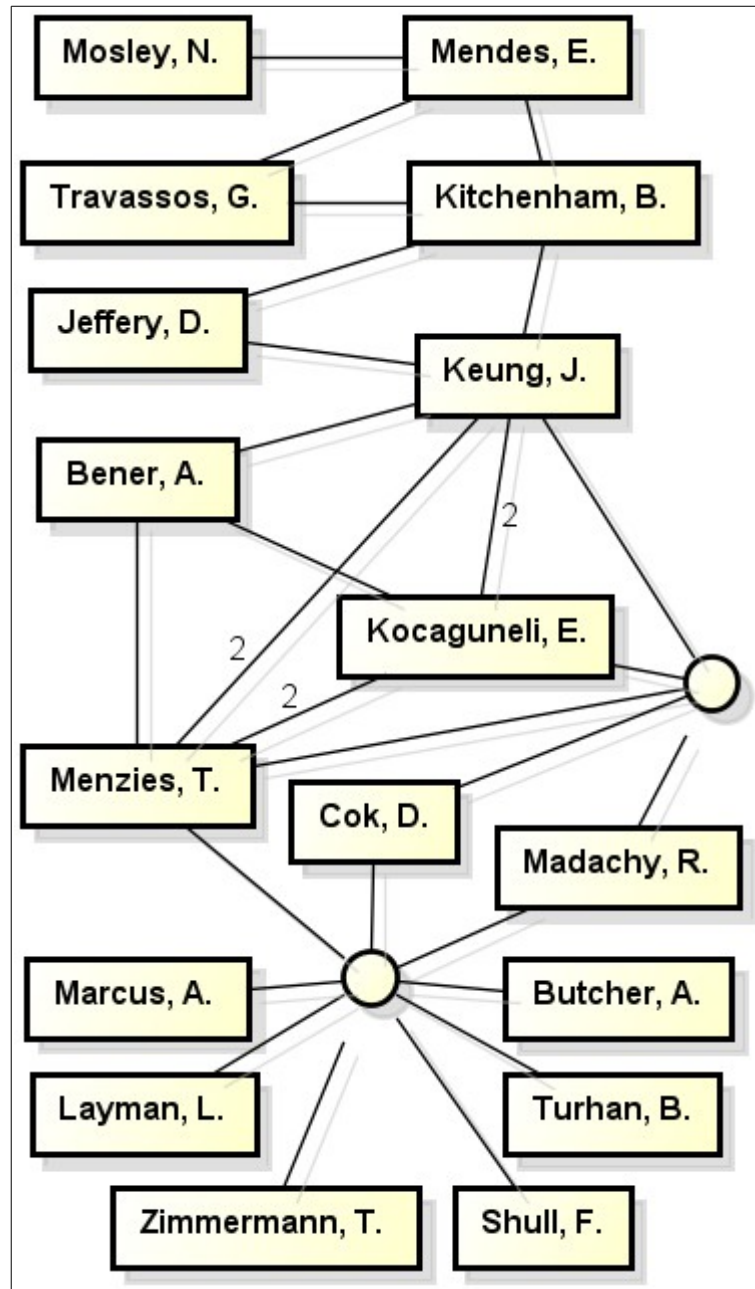


Figura 47 – Maior rede de cooperação entre autores.

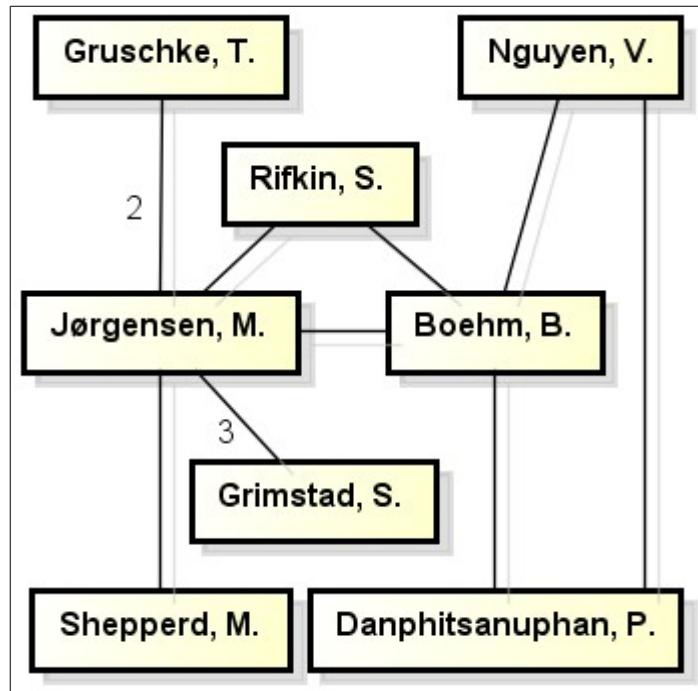


Figura 48 – Rede de cooperação entre autores de auto fator de publicação.

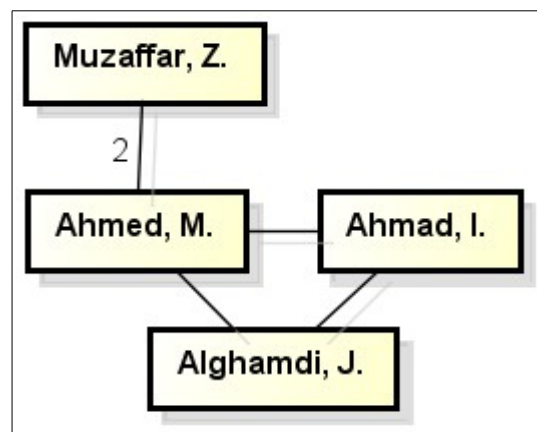


Figura 49 – Rede de cooperação influenciada principalmente por Moataz Ahmed.

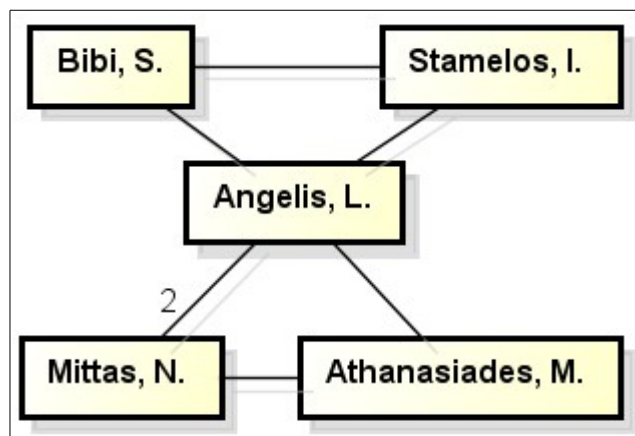


Figura 50 – Rede de cooperação influenciado principalmente por Lefteris Angelis.

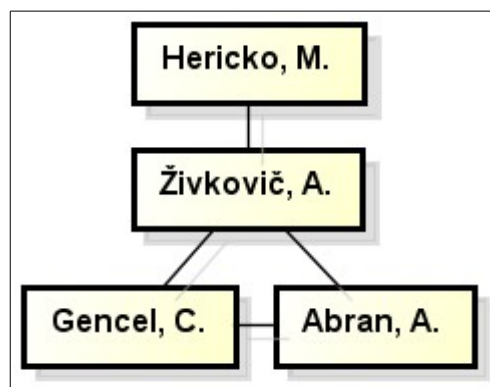


Figura 51 – Rede de cooperação influenciado principalmente por Aleš Živkovič.

A.1.2.3 QP3. Que Revistas Costumam Publicar na Área?

Pelo fato das revistas Information And Software Technology e IEEE Transactions on Software Engineering serem focadas na área de pesquisa em Engenharia de Software, obtiveram respectivamente, 25 (49,02%) e 15 (29,41%) artigos. Enquanto as outras três revistas não obtiveram nem 22% dos artigos selecionados juntas. Os resultados estão presentes na Tabela 13.

Tabela 13 – Número de artigos selecionados por revista.

Revista	Abreviatura	Número de Artigos	%
ACM Transactions On Software Engineering And Methodology	ACM TOSEM	3	5,88
Computer	Comp.	1	1,96
IEEE Software	IEEE SW	7	13,73
IEEE Transactions on Software Engineering	IEEE TSE	15	29,41
Information And Software Technology	IST	25	49,02

A.1.2.4 QP4. Quais São as Principais Abordagens para Estimativa de Esforço de Software?

Na revisão sistemática da literatura realizada por Jørgensen e Shepperd (2007), as abordagens foram separadas nas 12 categorias “Analogia”, “Árvores de Classificação e Regressão”, “Combinação de Estimativas”, “Julgamento de Especialista”, “Outros”, “Pontos de Função”, “Rede Bayesiana”, “Rede Neural”, “Regressão”, “Simulação”, “Teoria”, e “Work break-down”. Para comparação, a divisão das abordagens utilizada por Jørgensen e Shepperd (2007) foi seguida nesta revisão. Na Tabela 14, são apresentadas as contagens de artigos por abordagem.

Tabela 14 – Contagem e comparação de abordagens encontradas e comparação com os resultados encontrados por Jørgensen e Shepperd (2007).

Abordagem	Encontrados por Jørgensen e Shepperd (2007)	Encontrados por Jørgensen e Shepperd (2007) (%)	Encontrados entre os selecionados	Encontrados entre os selecionados (%)	Aumento Relativo (%)
Analogia	31	10	11	22	120
Árvores de Classificação e Regressão	14	5	12	24	380
Combinação de Estimativas	5	2	2	4	100
Julgamento de Especialista	46	15	19	37	147
Outros	25	8	4	8	0
Pontos de Função	68	22	11	22	0
Rede Bayesiana	7	2	3	6	200
Rede Neural	22	7	8	16	129
Regressão	148	49	19	37	-24
Simulação	10	3	1	2	-33
Teoria	39	13	2	4	-69
Work break-down	12	4	1	2	-50

As abordagens mais frequentes foram “Julgamento de Especialista” e “Regressão” com 19 artigos cada um, seguidos por “Árvores de Classificação e Regressão” com 12, e “Analogia” e “Pontos de Função” com 11 artigos cada um.

A frequência da abordagem “Julgamento de Especialista” foi claramente influenciada pelos estudos de Jørgensen, que publicou 9 dos 19 artigos.

A frequência da abordagem “Regressão” pode ser justificada pela simplicidade de aplicação desta abordagem e a capacidade de realizar variações de implementação.

Ao comparar os resultados encontrados com os resultados apresentados por Jørgensen e Shepperd (2007), foi possível encontrar a tendência de crescimento de interesse de pesquisa nas abordagens baseadas em “Analogia”, com 120% de aumento, “Árvores de Classificação e Regressão”, com 380% de aumento, “Combinação de Estimativas”, com 100% de aumento, “Julgamento de Especialista”,

com 147% de aumento, “Rede Bayesiana”, com 200% de aumento, e “Rede Neural”, com 129% de aumento.

A.1.2.5 QP5. Que Estudos Mostraram Melhoras Significativas na Predição de Tempo de Execução das Atividades de Software? E para Quais Abordagens?

Os artigos encontrados entre os selecionados que trazem resultados positivos aplicáveis estão descritos no Quadro 16. E, embora outros artigos como os de Kocaguneli et al. (2012a, 2012b) apresentem avanços em seus experimentos para multimétodos e métodos baseados em Analogia (introduzindo o método TEAK), estes não fornecem índices de erro, como Magnitude de Erro Relativo Médio (MMRE) ou Magnitude de Erro Relativo Mediano (MdMRE), e por compararem um conjunto muito restrito de métodos, se torna difícil julgar sua real contribuição.

Abordagem	Melhoria encontrada	Referência
Julgamento do Especialista	Melhores resultados são encontrados ao se remover informações irrelevantes a análise do requisito e ao se remover a descrição do requisito (como “nova funcionalidade” e “pequena extensão”) por influenciarem negativamente o estimador.	(Jørgensen e Grimstad 2012)
Julgamento do Especialista	Melhores resultados são encontrados ao se evitar realizar estimativas de requisitos similares um após a outra	(Jørgensen 2013b)
Ordinary Least-Squares Regression (OLS)	Melhores resultados são obtidos ao utilizar a transformação logarítmica dos elementos, cujo índice de erro Magnitude de Erro Relativo Mediano (MdMRE) demonstrou quedas entre 6% e 94%.	(Dejaeger et al. 2012)
Ordinary Least-Squares Regression (OLS)	Melhores resultados podem ser encontrados se os conjuntos de dados forem separados em grupos determinados por um valor de limite, como um indicador de acurácia. E um modelo <i>Least Squares Regression</i> (LSR) for criado para cada grupo. Atingindo melhoria de acurácia entre 34% e 42%, ao analisar Magnitude de Erro Relativo Média (MMRE), Magnitude de Erro Relativo Mediana (MdMRE), Magnitude de Erro Relativo a Estimativa Média (MMER), Magnitude de Erro Relativo a Estimativa Mediana (MdMER), e MMRE Balanceado (BMMRE).	(Seo et al. 2013)
Pontos de Função	A calibração de Pontos de Função, utilizando um modelo <i>neuro-fuzzy</i> , resulta em aumentos de acurácia das estimativas, pelo índice Magnitude de Erro Relativo Médio (MMRE), em 22%.	(Xia et al. 2008)
Support Vector Regression (SVR), MLP Neural Networks e M5P	Melhores resultados são obtidos ao aplicar um algoritmo genético para seleccionar as melhores atributos de entrada e os melhores parâmetros de uma técnica de aprendizagem de máquina.	(Oliveira et al. 2010)

Quadro 16 – Melhorias de estimativa identificadas nos artigos selecionados.

Especificamente para as abordagens baseadas em Julgamento de Especialista, os resultados dos experimentos apresentados no Quadro 17 resumizam o impacto de algumas variáveis no processo de estimativa de software. E, desta forma, ao controlar melhor estas variáveis, se obtém estimativas mais próximas do ideal.

Variável	Valor Mediano em Horas	Impacto	Referência
Requisito apresentado como pequena extensão.	85	Estimativa 21% menor do que a mais próxima da ideal.	(Jørgensen e Grimstad 2012)
Requisito apresentado como nova funcionalidade.	107		
Requisito apresentado com informações irrelevantes.	90	Estimativa 36% maior do que a mais próxima da ideal.	
Requisito apresentado sem informações irrelevantes.	66		
Requisito simples (de esforço menor que 20 homens/hora) apresentado após requisito complexo (de esforço maior que 180 homens/hora).	64	Estimativa 220% maior do que a ideal.	(Jørgensen 2013b)
Requisito complexo (de esforço maior que 180 homens/hora) apresentado após requisito simples (de esforço menor que 20 homens/hora).	150	Estimativa 17% menor do que a ideal.	
Requisito complexo (de 300 homens/hora) A.	220	Estimativa cresceu em 25% em comparação com requisito similar, na segunda estimativa.	
Requisito complexo (de 300 homens/hora) B, sendo B estimado com base em estimativa de A.	275		
Requisito complexo (de 300 homens/hora) B.	240	Estimativa cresceu em 33% em comparação com requisito similar, na segunda estimativa.	
Requisitos complexos (de 300 homens/hora) A, sendo A estimado com base em estimativa de B.	320		

Quadro 17 – Resultados de experimentos de impacto nos resultados de estimativas baseadas em julgamento do especialista.

A.1.2.6 QP6. Quais os Tipos de Atividades de Software que Divergem Mais Entre o Tempo Estimado e o Tempo Realizado?

Os artigos que discriminam a natureza das atividades como atributos para definição de complexidade são os descritos no Quadro 18. A pesquisa descrita por Menzies et al. (2013) divide as atividades em três categorias “New development”, “Enhancement” e “Redevelopment”, e apenas utiliza esta distinção como atributo para os métodos de estimativa baseados em aprendizado de máquina, sem discorrer sobre a influência da natureza sobre a realização da atividade, assim como

Huang et al. (2008). Porém, Lavazza et al. (2013) afirma que o esforço empregado na melhoria de um software é influenciado pelo tamanho do software original, diferente do esforço empregado em novo desenvolvimento, o que sugere que se utilizem dados do software original para realização da estimativa de atividades de melhoria.

Natureza da Atividade	Termo Original	Referência
Novo desenvolvimento	New development	(Menzies et al. 2013) (Huang et al. 2008) (Lavazza et al. 2013)
Melhoria	Enhancement	(Menzies et al. 2013) (Huang et al. 2008) (Lavazza et al. 2013)
Redesenvolvimento	Redevelopment	(Menzies et al. 2013) (Huang et al. 2008)
Pequena melhoria	Minor extension	(Jørgensen e Grimstad 2012)
Nova funcionalidade	New functionality	(Jørgensen e Grimstad 2012)

Quadro 18 – Naturezas das atividades discutidas entre os artigos selecionados.

Já para Jørgensen e Grimstad (2012), a natureza da atividade teve a sua grande influência na estimativa baseada em Julgamento de Especialista discutida e, em seus experimentos, cujos resultados estão apresentados no Quadro 19, apresentaram uma redução de 21% do esforço estimado para realização de atividades de desenvolvimento de software pela simples mudança deste atributo.

Natureza da Atividade	Termo Original	Mediana das Horas de Trabalho Estimadas
Pequena melhoria	Minor extension	85
Nova funcionalidade	New functionality	107

Quadro 19 – Horas de trabalho estimadas por natureza.

Os artigos discutidos nesta revisão não foram suficientes para responder esta questão, o que aponta para a falta de interesse dos pesquisadores em lidar

com esta característica das atividades de desenvolvimento de software.

A.1.3 Discussão da Revisão Sistemática da Literatura em Estimativa de Esforço de Software

Após analisar os artigos selecionados, ficou claro o interesse em estudos envolvendo julgamento de especialista. Segundo Jørgensen e Shepperd (2007), pesquisas que abordam julgamento de especialista são muito esparsas e merecem mais esforço de pesquisa. E, no estudo realizado por Dejaeger et al. (2012), os conjuntos de dados foram insuficientes para cobrir este contexto de julgamento de especialista. Já o estudo realizado por Kocaguneli et al. (2012a) não cobriu os efeitos de remoção de ruído e *outliers*, que podem impactar nos resultados.

Keung et al. (2008) afirma que métodos baseados em analogia automatizada são atualmente alternativas populares a abordagens de estimativa de esforço de software, comparados com sistema COCOMO (Benediktsson et al. 2003) ou outros métodos de construção de modelo baseado em regressão linear. Porém não há consenso de sua eficiência.

Symons (2012) apresenta a relação entre esforço e duração de uma atividade para argumentar que há o crescimento do esforço quando há necessidade de diminuição de prazo, o que ocorre com frequência na indústria de software. Sua pesquisa utilizou 16 subconjuntos de dados públicos, analisando 600 projetos reais de desenvolvimento de software. Os projetos foram agrupados por linguagem de programação primária e separados em projetos novos e projetos de melhoria.

Symons (2012) afirma a necessidade de discriminação do projeto entre projetos novos e de melhorias para melhores resultados na estimativa de software. Porém, em seu artigo, não discrimina os projetos por época em que foram realizados, metodologia de desenvolvimento, tamanho do projeto, ou tamanho da equipe, o que dentre outros fatores pode causar grande impacto na relação entre esforço e duração da atividade. E como os resultados de seu experimento diferem bastante entre o uso do primeiro conjunto de dados e do segundo, as conclusões desta pesquisa podem não ser aplicáveis para outros casos.

Suelmann (2013), em sua simulação de Monte Carlo, contradiz os resultados

encontrados por Symons (2012). O primeiro autor expõe o erro de derivação ocorrido no artigo original através de cálculos matemáticos. E discorda que o modelo possa refletir a realidade, já que a duração do projeto afeta o esforço e vice-versa.

Kocaguneli et al. (2012a), em seu artigo, apresenta o conceito de multimétodos como combinação dos métodos tradicionais de estimativa de software. Também realizam diversas comparações de performance de métodos e multimétodos, utilizando variações de preprocessadores, de métodos de discretização e de seleção de atributos. Aplicam Árvores de Classificação e Regressão (CART), Redes Neurais, métodos baseados em regressão, e métodos de aprendizagem baseada em instância. E nos testes realizados, CART se mostrou o método individual com melhor desempenho. Na maioria dos casos, a combinação de métodos obteve melhores resultados que os métodos individuais. Porém, métodos baseados em outras abordagens teriam de ser também comparados para suportar melhor a conclusão.

Kocaguneli et al. (2013) propõe um novo método de estimativa de software chamado QUICK. Este utiliza a distância euclidiana entre as instâncias dos conjuntos de dados. Distância essa calculada por uma matriz de atributos selecionados sem supervisão. Em seus experimentos, utilizaram 18 conjuntos de dados coletadas através da abordagem COCOMO. Utilizaram como métodos de estimativa para comparação CART e vizinho mais próximo (passiveNN). Antes da comparação dos métodos, os dados para análise foram normalizados, alguns discretizados e seus atributos selecionados. Utilizaram como métricas de comparação *absolute residual* (AR), *magnitude of error relative* (MER), *mean magnitude of error relative* (MMER), *median magnitude of error relative* (MdMER), porcentagem de predições falhando em 25 por cento do valor real (PRED(25)), *mean balanced relative error* (MBRE), e *mean inverted balanced relative error* (MIBRE). CART obteve um desempenho melhor em 22% dos casos que QUICK e nos outros 78% não houve diferença significativa. Já o passiveNN obteve um desempenho melhor em 17% dos casos e QUICK em 5% dos casos. E embora o método desenvolvido não tenha superado os métodos comparados, se mostrou de simples implementação e possivelmente adequado para casos em que não se conhece a natureza dos conjuntos de dados por utilizar de seleção de atributos automática. Porém não foram realizados testes comparativos com outros métodos, o que traz dúvidas a sua real eficácia. Assim como em Kocaguneli et al. (2012a), os

experimentos apontam o CART como o método com melhor desempenho.

Já Mittas e Angelis (2013), em seu artigo, no qual apresentam uma metodologia estatística para comparação de modelos de predição para estimativa de software. Utilizam 6 conjuntos de dados para comparar 11 modelos de predição, com método de validação *k-Fold Cross-Validation*, e apresentam como método com melhor desempenho o *Ordinary Least-Squares Regression (OLS)* e o *Classification And Regression Tree (CART)* como o com menor desempenho para realização de estimativas de esforço, discordando de Kocaguneli et al. (2013), que apresentou CART como com melhor desempenho. Porém a falta do uso de *Median of the Magnitude of Relative Error (MdmRE)* como medida de taxa de erro, e a não utilização de *Leave-One-Out Cross-Validation* como estratégia de validação, podem trazer resultados diferentes, como encontrado em (Kocaguneli et al. 2012b).

Para Dejaeger et al. (2012), que em seu artigo compara a performance das abordagens de estimativa de software baseadas em mineração de dados “*Ordinary least squares regression (OLS)*”, “*OLS regression with log transformation (LMS)*”, “*OLS regression with Box Cox (BC) transformation*”, “*Robust regression*”, “*Ridge regression*”, “*Least median squares regression*”, “*Multivariate adaptive regression splines (MARS)*”, “*Classification and regression tree (CART)*”, “*Model tree*”, “*Multilayered perceptron neural network*”, “*Radial basis function networks*”, “*Case-based reasoning*”, e “*Least squares support vector machines*”, a técnica que obteve o melhor desempenho foi a OLS, por obter os melhores resultados em sete dos nove casos, concordando com os resultados encontrados por Mittas e Angelis (2013) em seus experimentos. Como métricas de comparação das técnicas de estimativa, utilizaram MdmRE, Pred(25) e correlação.

Dejaeger et al. (2012) afirmam que técnicas de aprendizagem de máquina funcionam melhor com conjuntos de dados pequenos e relevantes, do que com conjuntos de dados grandes e com ruídos, além de indicar atributos como linguagem de programação, tamanho da equipe e setor da companhia por se provarem bons atributos para estimativa de esforço. Dejaeger et al. (2012) sugerem também que técnicas de mineração de dados podem contribuir para o processo de estimativa de esforço como um complemento ao julgamento do especialista, determinando um multimétodo, assim como explorado por Kocaguneli et al. (2012a).

Mendes e Mosley (2008), em seu artigo compara a performance de Redes Bayesianas Orientadas a Dados, Redes Bayesianas Híbridas, Analogia, Regressão,

Esforço Médio e Esforço Mediano. Utilizando o conjunto de dados Tukutuku e comparando os resultados encontrados utilizando as métricas, Magnitude de Erro Relativo Média (MMRE), Magnitude de Erro Relativo Mediano (MdMRE), porcentagem de previsões falhando em 25% do valor real (PRED(25)), EMRE Médio (MEMRE) e EMRE Mediano (MdEMRE). E assim como Dejaeger et al. (2012) e Mittas e Angelis (2013), obteve como a abordagem mais eficiente a Regressão, seguido por Redes Bayesianas Híbridas e Analogia.

Hearty et al. (2009) apresentam a aplicação de técnica de Redes Bayesianas para previsão de velocidade da execução de iterações para projetos realizados seguindo a metodologia de desenvolvimento ágil *Extreme Programming*. Embora não tenham sido realizados testes com conjuntos de dados públicos, pelo conjunto de dados não ser acessível a outros autores e não terem realizado comparações com outros métodos de estimativa, não há como saber se haveriam métodos com melhor desempenho para o caso abordado. E segundo o que autores como Mendes e Mosley (2008) afirmam, poderiam ser encontrados melhores resultados ao utilizarem outros métodos mais apropriados para problemas desta natureza, como métodos baseados em Regressão.

Para o terceiro método com melhor desempenho, segundo Mittas e Angelis (2013), métodos baseados em Analogia carecem de base estatística (Keung et al. 2008). E por isso Keung et al. (2008), em seu artigo, propôs uma extensão de métodos baseados em Analogia, incluindo inferência estatística baseada no método de Mantel. E para atestar sua eficácia, os autores aplicaram o método de Mantel para prover medidas quantitativas ao método baseado em analogia, permitindo verificar a aplicabilidade de Analogia para um determinado conjunto de dados. Além de definir uma base estatística para Analogia e prover um mecanismo para seleção de atributos.

Kocaguneli et al. (2012b), não satisfeitos com a performance de métodos baseados em Analogia, propõem a heurística TEAK, que baseado no princípio do caminho fácil, remove todos os casos cuja técnica de estimativa a ser utilizada teria dificuldades para resolver, de modo a apenas deixar os casos que se encaixam bem na lógica da técnica a ser utilizada. Para atestar esta heurística, foram realizadas comparações utilizando 10 conjuntos de dados públicos, com as técnicas de redes neurais, vizinho mais próximo e regressão linear. Comparando as técnicas mais eficientes, através das métricas Erro Relativo Médio, PRED(25), e Resíduo Absoluto.

Porém não utilizou a métrica Erro Relativo Mediano, o que poderia trazer resultados mais significativos. E não utilizou o conceito da heurística TEAK para outras abordagens, o que poderia ser bastante interessante, especialmente com abordagens de alto custo computacional, por diminuir o conjunto de dados.

Jørgensen e Shepperd (2007) apresentam, em seu artigo, uma revisão sistemática da literatura em estimativa de custo de software para artigos de data de publicação até 2004, na qual utilizaram métodos de pesquisa automática e manual para encontrar 304 artigos de 76 revistas, e os analisaram com o objetivo principal de encontrar os problemas mais comuns. Foram classificados os artigos por tópico de pesquisa, abordagem de estimativa, abordagem de pesquisa, contexto de estudo e uso de conjuntos públicos de dados. E concluíram que pesquisas publicadas sobre Julgamento de Especialista não eram suficientes na época. E embora este artigo compare um grande número de artigos, o faz para um contexto de pesquisa bastante diferente do atual.

Jørgensen et al. (2009) apresentam uma discussão entre os benefícios do uso de métodos de estimativa baseados em Julgamento do Especialista e benefícios do uso de métodos baseados em modelo. Em seu artigo são levantados os problemas de cada um como a necessidade de manter dados históricos íntegros para alimentar os modelos e a falta de visão de negócio destes modelos e a dificuldade humana de lidar com seus vieses individuais e sua dificuldade de realizar estimativas objetivas ou de aprender com os erros passados. Aprendizado este que Jørgensen e Gruschke (2009) afirmam que podem sim diminuir a influência dos fatores negativos na estimativa, sugerindo a revisão de todas as estimativas, forçando o estimador a refletir sobre suas estimativas antigas e aprender com os erros.

Jørgensen e Grimstad (2008), em seu artigo, afirmam que o impacto de informações irrelevantes afeta significativamente a acurácia da estimativa de software. Tendo testado suas hipóteses em sua pesquisa, na qual participaram 170 profissionais, distribuídos aleatoriamente entre 4 grupos, cujos três primeiros grupos receberam uma grande quantidade de informações irrelevantes, enquanto que o quarto grupo apenas recebeu informações relevantes a realização da estimativa. No segundo grupo, os profissionais foram orientados a destacar as informações relevantes. No terceiro grupo, os profissionais foram orientados a riscar as informações irrelevantes.

Jørgensen e Grimstad (2008) observaram que destacar informações relevantes ou riscar informações irrelevantes não surtiu o efeito desejado, além de que pesquisa a presença das informações irrelevantes causou aumento de 40% a 60% em homens-horas estimados. Porém não levaram em consideração aspectos pessoais dos participantes como sexo, experiência, ou aspectos culturais, o que poderia ter enriquecido a pesquisa e/ou levantado questões.

Jørgensen e Grimstad (2011) retomaram a sua pesquisa (Jørgensen e Grimstad 2008), porém conduzindo experimentos com documentos de análise com diferentes quantidades de informações irrelevantes em ambientes de laboratório (acadêmico) e de campo (indústria). Apresentam algumas diferenças encontradas entre os resultados obtidos em laboratório e em campo, discutindo também algumas dificuldades enfrentadas na realização destes experimentos.

Outros vieses causados por informações irrelevantes foram levantadas por Jørgensen (2013b), apresentando uma pesquisa sobre a influência que estimadores sofrem ao estimar mais de um projeto ou atividade, por imaginar relações inconscientemente entre projetos e/ou atividades independentes.

Embora a estimativa seja uma atividade que requeira conhecimento e familiaridade com o ambiente de produção, há sempre vieses humanos. Como, por exemplo, uma atividade, apenas tendo sua classificação alterada de “pequena alteração” para “nova funcionalidade” geraria a expectativa de demandar mais esforço (Jørgensen 2013b). E para diminuir a influência dos vieses humanos, entre outras considerações, o autor indica a combinação de várias estimativas, sendo realizadas por mais de um estimador e por mais de um método.

Basten e Sunyaev (2011), assim como Jørgensen (2013b), apresentam uma lista de fatores que podem causar vieses na realização de estimativas de software. Estes fatores foram encontrados ao realizar a análise de 50 pesquisas empíricas publicadas desde 1990, identificando fatores que tendem a afetar a acurácia da estimativa de software. Agrupando os fatores por “Processo”, “Estimador”, “Projeto”, e “Contexto”. Basten e Sunyaev (2011), neste artigo, apenas auxiliam estimadores a diminuir o impacto dos vieses presentes no processo de estimativa de software, porém não aponta quais fatores impactam mais ou quais são os mais comuns, nem lista os artigos nos quais estes fatores foram encontrados.

Jørgensen e Grimstad (2012), realizaram uma pesquisa com 374 desenvolvedores de 6 países, conduzindo 3 experimentos com objetivo de relacionar

problemas e acurácia de estimativa e os atributos dos desenvolvedores estilo de pensamento, como se veem, nacionalidade, experiência, habilidade, educação, sexo e papel organizacional. Porém a pesquisa teve conclusão contraditória para a relação dos atributos de experiência, educação e papel organizacional e embora afirmem que vieses podem depender da cultura ou de outros fatores contextuais, não indicam quais fatores ou quanto estes influenciam.

Nan e Harter (2009), diferente de Jørgensen e Grimstad (2012), focam na pressão exercida pelo cronograma e pelo orçamento. Tendo como base a definição de pressão como a relação entre “tempo/orçamento estimado” e “tempo/orçamento negociado”, da análise de 14 artigos de 1974 a 2005, e de um conjunto de dados privado de uma empresa de tecnologia de 25 bilhões de dólares de receita anual, foi analisada a relação encontrada entre cronograma e pressão e orçamento e pressão e se mostraram não lineares e benéficos até 10% de pressão, porém acima deste valor se mostraram prejudiciais. Pelos testes serem de simples implementação e utilizar apenas um conjunto de dados privado, se torna interessante realizar testes em conjuntos de dados públicos para, além de verificar as conclusões dadas, cruzar com as técnicas de estimativa de software para verificar a relação entre as duas.

Kitchenham et al. (2007), em sua revisão sistemática da literatura, teve como objetivo determinar se empresas podem se beneficiar de modelos desenvolvidos por outras empresas para estimativa de software. Realizada com base em dez artigos, entre 1999 e 2005, esta pesquisa se apresenta como inconclusiva e apresenta a dificuldade em compatibilizar resultados de estudos comparativos, já que estes são realizados sem preocupação quanto a padronização de procedimentos de experimentação.

Menzies et al. (2013), aborda o mesmo tema que Kitchenham et al. (2007), da discussão do dilema entre a aplicação de conjuntos de dados e técnicas já estabelecidos e a mineração de dados e implementação das técnicas específicas para seu contexto de aplicação. Esta pesquisa, através de uma revisão da literatura focando em estimativa de software e predição de defeitos, apresentou a análise de 4 e 28 artigos, para respectivamente, estimativa de software e predição de defeitos. Para os experimentos, foram agrupadas as instâncias pelas dimensões de maior variabilidade para analisá-las em um contexto de grupo, como meio-termo a contexto local e global. Os resultados encontrados foram bastante positivos para o uso de instâncias de contexto de grupo, comparado aos outros contextos. O que

argumenta para a conclusão de que métodos baseados em modelos, que aprendem não somente do conjunto de dados do contexto no qual serão aplicados, mas também de conjuntos de dados de contextos similares, obtêm melhores resultados.

Jørgensen (2009), em seu artigo apresenta várias recomendações para o profissional que avalia os licitantes das empresas, com o objetivo de evitar decidir por empresas cujo orçamento seja superotimista. E assim como em “*Guidelines for Software Development Effort Estimation*” (Basten e Sunyaev 2011), o autor apenas descreve alguns fatores que podem levar a decidir pela licitante não competente ou cujo orçamento superotimista levará a má qualidade de software, mas sem apontar os problemas mais comuns ou seu impacto no processo de avaliação de orçamento.

Buglione e Ebert (2011) apresentam uma lista de ferramentas para realização correta, do ponto de vista gerencial, da estimativa de software. Em sua pesquisa foram reunidas ferramentas de auxílio a gerentes de projetos para realização de estimativa de esforço, gerência de requisitos e metas, pois frequentemente pessoas confundem estimativas com necessidades, planejando atividades não com base no realizável, mas sim nas expectativas de clientes.

A.1.4 Conclusão da Revisão Sistemática da Literatura em Estimativa de Esforço de Software

Esta revisão sistemática da literatura apresenta, ao analisar artigos entre 2008 e 2013, o crescimento do interesse em pesquisas em estimativa de esforço de software. O protocolo elaborado foi seguido como definido, porém para as questões de pesquisa 5 e 6, os artigos selecionados não foram suficientes para dar suporte a melhores conclusões e, portanto, se tornam necessárias pesquisas mais específicas.

A estimativa de esforço de software detém uma etapa importante da análise de viabilidade de projetos e sua acurácia ainda carece de melhoria, pois como exibido por Gruschke e Jørgensen (2008) em seus experimentos, estimativas baseadas em Julgamento do Especialista tiveram diferença entre esforço estimado mediano e esforço real entre 8% e 63%, o que aponta a instabilidade da técnica. Porém mesmo a melhor das técnicas baseadas em aprendizado de máquina

apresentou diferença entre esforço estimado mediano e esforço real entre 27,5% e 60,8%, o que também se mostra insatisfatório (Dejaeger et al. 2012).

Nos artigos selecionados foi clara a influência do autor Magne Jørgensen, pelo grande número de artigos, especialmente para Julgamento de Especialista, tema recorrente para este autor. Embora diferente de Jørgensen que costuma publicar individualmente ou entre poucos colegas, Tim Menzies influenciou um número maior de autores por trabalhar diretamente ou indiretamente com eles. Já a revista que causou maior influência entre os artigos selecionados foi a *Information And Software Technology*, que por focar mais em Engenharia de Software, demonstrou ser a preferida para publicações na área de Estimativa de Software.

Como apenas constaram cinco revistas dentre os artigos selecionados, se pode sugerir que estas revistas possam causar influência no resultado desta revisão sistemática da literatura, já que podem favorecer autores e/ou abordagens. E por conta disso, se recomenda em futuros estudos expandir o número de revistas aceitas, diminuindo o fator de impacto exigido.

Entre os métodos que mais estão ganhando interesse na área estão “Árvores de Classificação e Regressão”, “Julgamento de Especialista” e “Rede Bayesiana”. Curiosamente, muito pouco foi encontrado sobre as características, natureza e momento de execução das atividades estimadas, o que levanta a possibilidade a ser descoberta em estudos futuros de serem informações relevantes ignoradas ou irrelevantes corretamente descartadas.

Pouco se avançou nestes últimos anos quanto a acurácia dos métodos, porém pesquisas comparativas têm se mostrado frequentes, embora geralmente incompatíveis por usarem métodos de validação e métricas de comparação não apropriados, incompatíveis ou omissos. E, embora não haja consenso sobre quais abordagens se comportam melhor para esta tarefa, algumas delas se destacam como *Ordinary Least-Squares Regression*, *Robust M-estimator Regression*, *Least Median Squares Regression* e *Least Trimmed Squares Regression* (Mittas e Angelis 2013), além de abordagens baseadas em Analogia e *Classification And Regression Tree* (Kocaguneli et al. 2012a).

A.2 REVISÃO DO ESTADO DA ARTE EM *PLANNING POKER*

A pesquisa realizada focou em *Planning Poker*, e teve sua realização necessária pelo tema não ter sido anteriormente explorado na Revisão Sistemática da Literatura em Estimativa de Esforço de Software.

A.2.1 Protocolo de Pesquisa de Artigos da Literatura

Diferente da revisão sistemática da literatura, que pretende responder a questões relativas ao tema, esta pesquisa apenas pretende descrever o estado da arte, sem necessariamente gerar resultados quantitativos.

A.2.1.1 Estratégia de Pesquisa de Artigos da Literatura

Por conta do grande volume de artigos e da disponibilidade destes, foram excluídos os artigos de patentes e os apenas disponíveis em livros. A busca por artigos foi realizada filtrando-se os artigos por ano de publicação, tomando-se os anos de 2008 a 2013, e utilizando a palavra-chave "*Planning Poker*", para assim obter resultados pertinentes ao tema, independente da revista. Após este filtro inicial, os artigos encontrados foram filtrados manualmente por título, resumo e por conteúdo do artigo. Esta estratégia de pesquisa difere da utilizada na revisão sistemática da literatura por focar em um tema relativamente pouco explorado, e por isso disperso entre diversas revistas e conferências.

A.2.1.1.1 Seleção de Bases de Dados

Algumas bases de dados de artigos podem influenciar o conjunto de artigos encontrados por beneficiar um formato, um método de estudo ou uma subárea de pesquisa. E, para diminuir esta possível influência, quatro bases de dados bem-

conceituadas pela comunidade foram selecionadas:

- *ACM Digital Library*.
- *IEEE Xplore*.
- *Scopus*.
- *Web Of Science*.

A.2.1.2 Realização da Pesquisa de Artigos da Literatura

Com o protocolo definido, a pesquisa pode ser realizada. Este capítulo do artigo descreve a realização da pesquisa.

A.2.1.2.1 Seleção de Artigos para Inclusão

A Tabela 15 descreve o número de artigos de cada estágio do processo de seleção.

Tabela 15 – Horas de trabalho estimadas por natureza.

Estágio	Artigos	Adicionados	Total
Estágio 1: filtro por título	334	0	334
Estágio 2: filtro por resumo	98	0	98
Estágio 3: filtro por conteúdo do artigo	37	0	37
Final: artigos selecionados	20	0	20

A.2.1.2.2 Verificação da Seleção

Durante o primeiro estágio de seleção, pela quantidade de artigos a filtrar manualmente, foram realizadas buscas manuais por dois dos autores deste artigo.

As diferenças entre os resultados das buscas foram discutidas. Os resultados apresentados neste artigo foram analisados e acordados entre os autores. Apenas após a conclusão do segundo estágio de seleção, foram removidas as ocorrências repetidas nos resultados da busca.

A.2.1.2.3 Desvios do Protocolo

Nenhum desvio do protocolo foi necessário para conclusão da pesquisa.

A.2.1.2.4 Artigos Seleccionados

Os dados dos artigos seleccionados são apresentados no Quadro 12.

Revista ou Conferência	Autores	Ano	Título	Citação
IEEE Transactions on Education	Mahnič, V.	2012	A Capstone Course on Agile Software Development Using Scrum	(Mahnič 2012)
IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)	Nguyen-Cong, D.; Tran-Cao, D.	2013	A Review of Effort Estimation Studies in Agile, Iterative and Incremental Software Development	(Nguyen-Cong e Tran-Cao 2013)
IEEE 8th International Conference on Global Software Engineering Workshops (ICGSEW)	Kompella, L.	2013	Advancement of Decision-Making in Agile Projects by Applying Logistic Regression on Estimates	(Kompella 2013)
Proceedings of the 30th ACM international conference on Design of communication	McNely, B. J.; Gestwicki, P.; Burke, A.; Gelms, B.	2012	Articulating Everyday Actions: An Activity Theoretical Approach to Scrum	(McNely et al. 2012)
Information and Software Technology	Moe, N. B.; Aurum, A.; Dybå, T.	2012	Challenges of Shared Decision-Making: A Multiple Case Study of Agile Software Development	(Moe et al. 2012)
International Conference on Software and System Process (ICSSP)	Aroonvatanaporn, P.; Koolmanojwong, S.; Boehm, B.	2012	COTIPMO: A COConstructive Team Improvement Process Model	(Aroonvatanaporn et al. 2012)
35th International Conference on Software Engineering (ICSE)	Dagnino, A.	2013	Estimating Software-Intensive Projects in The Absence of Historical Data	(Dagnino 2013)
20th Asia-Pacific Software Engineering Conference (APSEC)	Raith, F.; Richter, I.; Lindermeier, R.; Klinker G.	2013	Identification of Inaccurate Effort Estimates in Agile Software Development	(Raith et al. 2013)

Quadro 20 – Lista de artigos selecionados.

Revista ou Conferência	Autores	Ano	Título	Citação
Software Quality Journal	Lagerström, R.; Württemberg, L. M.; Holm, H.; Luczak, O.	2012	Identifying Factors Affecting Software Development Cost and Productivity	(Lagerström et al. 2012)
Proceedings of the 33rd International Conference on Software Engineering	Boehm, B.; Valerdi, R.	2011	Impact of Software Resource Estimation Research on Practice: A Preliminary Report on Achievements, Synergies, and Challenges	(Boehm e Valerdi 2011)
International Journal of Approximate Reasoning	Guarino, S. L.; Pfautz, J. D.; Cox, Z.; Roth, E.	2009	Modeling Human Reasoning About Meta-information	(Guarino et al. 2009)
Journal of Systems and Software	Mahnič, V.; Hovelja, T.	2012	On Using Planning Poker for Estimating User Stories	(Mahnič e Hovelja 2012)
Proceedings of the CUBE International Information Technology Conference	Choudhari, J.; Suman, U.	2012	Phase Wise Effort Estimation For Software Maintenance: An Extended SMEEM Model	(Choudhari e Suman 2012)
43rd Hawaii International Conference on System Sciences (HICSS)	Heikkilä, V.; Jadallah, A.; Rautiainen, K.; Ruhe, G.	2010	Rigorous Support for Flexible Planning of Product Releases: A Stakeholder-centric Approach and Its Initial Evaluation	(Heikkilä et al. 2010)
International Symposium on Empirical Software Engineering and Measurement	Williams, L.; Brown, G.; Meltzer, A.; Nagappan, N.	2011	Scrum + Engineering Practices: Experiences of Three Microsoft Teams	(Williams et al. 2011)
Information and Software Technology	Miranda, E.; Bourque, P.; Abran, A.	2009	Sizing User Stories Using Paired Comparisons	(Miranda et al. 2009)

Quadro 20 – Lista de artigos selecionados (continuação).

Revista ou Conferência	Autores	Ano	Título	Citação
Proceedings of the 2010 ICSE Workshop on Software Development Governance	Peixoto, C. E. L.; Audy, J. L. N.; Prikladnicki, R.	2010	The Importance of The Use of An Estimation Process	(Peixoto et al. 2010)
Information and Software Technology	Børte, K.; Ludvigsen, S. R.; Mørch, A. I.	2012	The Role of Social Interaction in Software Effort Estimation: Unpacking The "Magic Step" Between Reasoning and Decision-Making	(Børte et al. 2012)
Journal of Systems and Software	Moløkken-Østfold, K.; Haugen, N. C.; Benestad, H. C.	2008	Using Planning Poker for Combining Expert Estimates in Software Projects	(Moløkken-Østfold et al. 2008)
Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement	Desharnais, J.-M.; Buglione, L.; Kocatürk, B.	2011	Using the COSMIC Method to Estimate Agile User Stories	(Desharnais et al. 2011)

Quadro 20 – Lista de artigos selecionados (continuação).

A.2.2 Introdução a *Planning Poker*

O método *Planning Poker* é considerado uma técnica de estimativa *top-down* baseado em julgamento de especialista, útil nos estágios iniciais do desenvolvimento, quando se estima o tamanho do software e o esforço é calculado baseado na velocidade do time de desenvolvimento (Dagnino 2013).

Segundo Raith et al. (2013), o fluxo de execução do *Planning Poker*, exibido na Figura 52, envolve todos os membros do time de desenvolvimento como estimadores, um moderador (*Scrum Master*, por exemplo) e, opcionalmente, um representante do cliente.

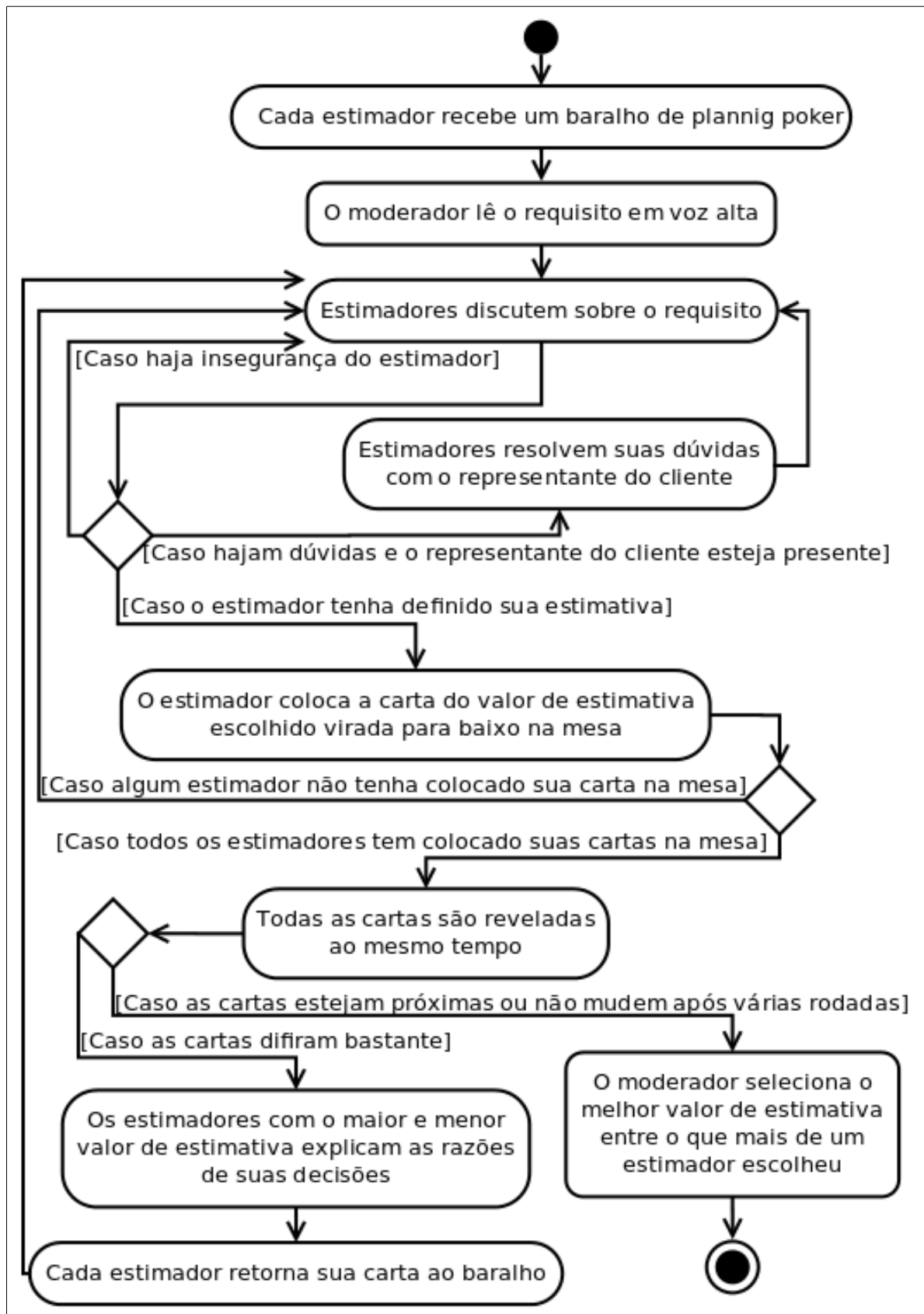


Figura 52 – Diagrama do fluxo de execução do *Planning Poker*..

A.2.3 Discussão da Revisão do Estado da Arte em *Planning Poker*

Esta técnica de estimativa de esforço de software se destaca por utilizar tanto o conhecimento técnico de desenvolvimento de software, como a capacidade de prever o software ainda a ser desenvolvido (Børte et al. 2012).

Planning Poker é uma técnica de estimativa baseado em julgamento do especialista bastante utilizado em métodos ágeis, especialmente Scrum e *Extreme Programming* (Mahnič e Hovelja 2012). Foi introduzido por Grenning (2002) e popularizado por Cohn (2005).

Outro fator positivo do uso da técnica de estimativa *Planning Poker* é que diversos especialistas compartilham diferentes ideias e perspectivas, sendo realmente útil para pensar nas diferentes condições que podem ocorrer durante a implementação dos requisitos (Desharnais et al. 2011). É diferente de outras técnicas de estimativa, pelo processo do *Planning Poker* ser basicamente um jogo, não se torna cansativo (Desharnais et al. 2011).

Segundo Williams et al. (2011), *Planning Poker* provê um meio estruturado para:

- Obter uma compreensão entre todos os participantes.
- Expor suposições ocultas de aspectos técnicos de implementação e verificação.
- Discutir as implicações da implementação do requisito no sistema.
- Descobrir e resolver ambiguidades através de perspectivas divergentes sobre o requisito.
- Expor alternativas para implementação do requisito.

Planning Poker é visto como uma técnica que provê melhores estimativas, comparadas a estimativas individuais, ao permitir que todos os estimadores participem, independente de sua capacidade de persuasão (Mahnič e Hovelja 2012), além de incentivar a interação social durante a construção de estimativas (Børte et al. 2012).

Segundo Børte et al. (2012), *Planning Poker* é interessante por:

- Ser organizada de modo a permitir que todos participem através de suas cartas, independente da posição ou experiência.
- Requisitar justificativas para os participantes que estimarem com o maior

e o menor valor, como parte do processo de estimativa.

- Utilizar as cartas de jogo como artefatos de realização de estimativa.
- Permitir, de modo organizado, que especialistas de diferentes áreas estimem juntos, auxiliando a compreensão da solução de problemas altamente complexos.

Comparado também com a média das estimativas, o processo necessário para realização do *Planning Poker* oferece estimativas mais acuradas e menos otimistas. Além do fato de estimativas realizadas em grupo resultarem em estimativas mais acuradas que estimativas individuais (Raith et al. 2013).

O uso de *Planning Poker* reduz o otimismo nas estimativas quando comparadas a combinação das estimativas individuais (Moløkken-Østvold et al. 2008) (Raith et al. 2013) (Williams et al. 2011). Porém, quando realizados por participantes inexperientes, para requisitos semelhantes e valores extremos, tiveram seus desvios intensificados (Raith et al. 2013).

A interação cara a cara pode ser um fator que faz com que os participantes tomem mais cuidado com suas decisões (Moløkken-Østvold et al. 2008).

Em teoria, *Planning Poker* possui diversas características que deveriam proporcionar melhoras na acurácia das estimativas de esforço por combinar conhecimento de diversos profissionais, poderem ser realizadas com métodos de desenvolvimento iterativo, e por revelarem as estimativas ao mesmo tempo, reduzindo o viés social da comparação (Moløkken-Østvold et al. 2008).

Segundo Peixoto et al. (2010), 12% dos 551 profissionais que responderam sua pesquisa, responderam que utilizam *Planning Poker* como principal método de estimativa. Porém apenas três quartos destes aplicam o método de forma adequada.

Moløkken-Østvold et al. (2008), citam Moløkken-Østvold e Jørgensen (2004), onde estes descrevem que as estimativas em grupo realizadas após uma discussão não estruturada eram menos otimistas e mais realistas que estimativas individuais, realizadas antes da discussão, combinadas. A principal fonte dessa diminuição de otimismo aparentou ser a identificação de atividades adicionais e a percepção de que atividades eram mais complexas do que inicialmente pensado.

Børte et al. (2012) e Moløkken-Østvold et al. (2008), citam Haugen (2006), que comparou o uso de *Planning Poker* com a realização de estimativas em grupo não estruturado para investigar o impacto no desempenho da estimativa. Com um total de 101 requisitos estimados em ambiente de desenvolvimento XP (Extreme

Programming). Mostrando, com seus resultados, o aumento da acurácia de estimativas para atividades familiares, comparado ao grupo não estruturado. Porém, com impacto negativo ao estimar atividades não familiares.

Segundo Moløkken-Østfold et al. (2008), um problema frequentemente reportado na área de pesquisa é que experimentos tendem a investigar atividades e/ou resultados hipotéticos, e não atividades executadas realmente com seus resultados registrados.

Embora a técnica de estimativa *Planning Poker* não seja tão confiável, por se basear em estimativas humanas e por não utilizar de análises como medidas históricas ou analogia, uma abordagem para prevenir estes problemas é utilizar um repositório histórico (Desharnais et al. 2011).

Moe et al. (2012) aponta a falta de conhecimento relacionado ao requisito, falta de tempo para preparação e análise, e falta de comunicação entre os integrantes do grupo de estimadores como causadoras de estimativas que subestimam a complexidade da implementação do requisito em ambientes de decisão compartilhada.

Infelizmente não há posição clara da área de pesquisa sobre os valores das cartas de jogo, pois mesmo sendo definido por Grenning (2002) com os valores “1”, “2”, “3”, “5”, “7”, “10” e “infinitos” dias, Cohn (2005), que popularizou a técnica, definiu os valores como “0,5”, “1”, “2”, “3”, “5”, “8”, “13”, “20”, “40” e “100” dias. Já Dagnino (2013), definiu os valores como “1”, “2”, “3”, “5”, “8”, “13”, “20”, “30”, “40”, “50”, “70”, “90” e “100” dias no melhor caso, convertendo as respostas em valores menos otimistas. Grenning (2002) permite ainda o uso de duas cartas para definir valores intermediários. Além de definir que o requisito deverá ser quebrado caso a carta “infinito” seja jogada.

A.2.4 Conclusão da Revisão do Estado da Arte em *Planning Poker*

Há poucas pesquisas que verificam a melhora no uso de *Planning Poker*, em comparação a estimativas realizadas por um só profissional, porém há um consenso na área de que as características positivas do método superam as negativas.

Embora os valores das cartas de jogo ainda estejam em discussão, os

valores definidos por Cohn (2005) demonstraram ter maior aceitação por serem utilizados por Raith et al. (2013), Børte et al. (2012), e Mahnič e Hovelja (2012).

APÊNDICE B – Termo de Consentimento Livre e Esclarecido

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Eu _____, código de aluno nº _____, estou sendo convidado a participar de um estudo denominado: “Influência da Revisão de Atividades Executadas para Melhoria da Acurácia Utilizando Planning Poker”, cujo objetivo é verificar o impacto na acurácia da realização de estimativas de esforço de desenvolvimento de software, utilizando Planning Poker.

Sei que para o avanço da pesquisa a participação de voluntários é de fundamental importância. Caso aceite participar desta pesquisa eu, em conjunto de colegas da disciplina, realizarei a estimativa de esforço de atividades de desenvolvimento de software em uma ferramenta web e responderei a um questionário de avaliação de usabilidade desta ferramenta.

Estou ciente de que minha privacidade será respeitada, ou seja, meu nome, ou qualquer outro dado confidencial, será mantido em sigilo. A elaboração final dos dados será feita de maneira codificada, respeitando o imperativo ético da confidencialidade.

Estou ciente de que posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar, nem sofrer qualquer dano.

Os pesquisadores envolvidos com o referido projeto são, o professor Doutor Laudelino Cordeiro Bastos, a professora Doutora Maria Cláudia Figueiredo Pereira Emer, e o mestrando do Programa de Pós-Graduação em Computação Aplicada André Augusto Tissot da Universidade Tecnológica Federal do Paraná, com quem poderei manter contato pelos e-mails: bastos@dainf.ct.utfpr.edu.br, mclaudia@dainf.ct.utfpr.edu.br, e andre_a_tissot@yahoo.com.br. Estão garantidas todas as informações que eu queira saber antes, durante e depois do estudo.

Li, portanto, este termo, fui orientado quanto ao teor da pesquisa acima mencionada e compreendi a natureza e o objetivo do estudo do qual fui convidado a participar. Concordo, voluntariamente em participar desta pesquisa, sabendo que não receberei nem pagarei nenhum valor econômico por minha participação.

Assinatura do sujeito de pesquisa

André Augusto Tissot
Assinatura dos pesquisadores

Laudelino Cordeiro Bastos

Curitiba _____ de _____ de 2014

APÊNDICE C – Questionário de Avaliação da Ferramenta

Avaliação da Ferramenta:

Dados pessoais

Data de preenchimento: _____

Curso: _____

Nome: _____

Data de nascimento: _____

Gênero: Masculino Feminino

Teve contato com estimativas antes de utilizar esta ferramenta? Sim Não

Navegador utilizado e versão: _____

Sistema operacional utilizado e versão: _____

Avaliação de Usabilidade

1. Tempo de Carregamento: As páginas normalmente carregam rapidamente.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

2. Apresentação do Conteúdo: A ferramenta web tem propósito bem definido.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

3. Apresentação do Conteúdo: A ferramenta web tem tema bem definido e este é mantido em todas páginas.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

4. Precisão do Conteúdo: As informações apresentadas na ferramenta web são precisas.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

5. Precisão do Conteúdo: O objetivo da ferramenta foi alcançado.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

6. Interesse: O conteúdo da ferramenta é interessante para seu público-alvo.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

7. Ortografia e Gramática: Não há erros de escrita, pontuação ou gramática na ferramenta web.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

8. Navegação: Os links usados para navegação são claramente nomeados, localizados de forma consistente.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

9. Navegação: Ao utilizar a ferramenta web, o usuário sempre sabe em que página está.

- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

10. Navegação: Ao utilizar a ferramenta web, o usuário descobre facilmente onde clicar para chegar a página que deseja.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
11. Compatibilidade: A ferramenta web funcionou corretamente com o(s) navegador(es) utilizado(s).
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
12. Plano de Fundo: O plano de fundo é consistente entre páginas.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
13. Plano de Fundo: O plano de fundo incrementa o tema da ferramenta web.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
14. Plano de Fundo: O plano de fundo incrementa o propósito da ferramenta web.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
15. Plano de Fundo: O plano de fundo não atrapalha a legibilidade.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
16. Escolha de Cores: As cores escolhidas para o plano de fundo, fontes e links formam uma paleta de cores agradável, não atrapalham a legibilidade do conteúdo e são consistentes entre as páginas.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
17. Leiaute: A ferramenta web utiliza um leiaute atrativo, auxiliando a localização de todos os elementos importantes. Espaços em branco, elementos gráficos e alinhamento são utilizados para organizar os dados apresentados.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
18. Fontes: O tipo de fonte do texto é fácil de ler.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
19. Fontes: O tipo de fonte do texto é consistente entre páginas.
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.
20. Fontes: O tamanho da fonte do texto muda de acordo com sua importância (títulos, subtítulos, texto, etc...).
- Concordo plenamente.
 Concordo parcialmente.
 Indiferente.
 Discordo parcialmente.
 Discordo completamente.

21. Fontes: O uso de estilos de fonte (itálico, negrito, sublinhado) é consistente.

- Concordo plenamente.
- Concordo parcialmente.
- Indiferente.
- Discordo parcialmente.
- Discordo completamente.

22. Fontes: O uso de estilos de fonte (itálico, negrito, sublinhado) auxilia a legibilidade.

- Concordo plenamente.
- Concordo parcialmente.
- Indiferente.
- Discordo parcialmente.
- Discordo completamente.

Avaliação de Satisfação

1. Utilizar a ferramenta me auxiliou a compreender o processo de estimativa.

- Concordo plenamente.
- Concordo parcialmente.
- Indiferente.
- Discordo parcialmente.
- Discordo completamente.

2. Utilizar a ferramenta me auxiliou a compreender o *Planning Poker*.

- Concordo plenamente.
- Concordo parcialmente.
- Indiferente.
- Discordo parcialmente.
- Discordo completamente.

3. Tenho interesse em utilizar a ferramenta para melhorar minhas estimativas.

- Concordo plenamente.
- Concordo parcialmente.
- Indiferente.
- Discordo parcialmente.
- Discordo completamente.

4. Tenho interesse em utilizar a ferramenta para realizar estimativas de forma prática.

- Concordo plenamente.
- Concordo parcialmente.
- Indiferente.
- Discordo parcialmente.
- Discordo completamente.